

For Reference

NOT TO BE TAKEN FROM THIS ROOM

Ex LIBRIS
UNIVERSITATIS
ALBERTAENSIS



THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Henry Y. Lieuson
TITLE OF THESIS EXPERIMENTAL EVALUATION OF SELF-TUNING
 CONTROL OF A BINARY DISTILLATION COLUMN
DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE
YEAR THIS DEGREE GRANTED Fall 1980

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission. D

THE UNIVERSITY OF ALBERTA

EXPERIMENTAL EVALUATION OF SELF-TUNING CONTROL OF A BINARY
DISTILLATION COLUMN

by

Henry Y. Lieuson



A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE

IN

PROCESS CONTROL

CHEMICAL ENGINEERING

EDMONTON, ALBERTA

Fall 1980

THE UNIVERSITY OF ALBERTA
FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled EXPERIMENTAL EVALUATION OF SELF-TUNING CONTROL OF A BINARY DISTILLATION COLUMN submitted by Henry Y. Lieuson, B.ApSc., in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE

ABSTRACT

This work concerns the experimental evaluation of the self-tuning controller (STC) on a pilot scale distillation column. The evaluation of the STC is done by comparing the performance of the STC with that of well tune proportional-integral-derivative (PID) controllers.

The ST and PID controllers are used to control the terminal compositions of the column when it is subjected to step changes in the feed flow rate. The controllers are tested in the following configurations: top composition control, bottom composition control, and simultaneous compositions control. For the self-tuning controller, dual ends control is performed using both the multiloop and multivariable arrangements. The performance of the controllers is quantified by considering the sum of the absolute error values over a specified time period after the feed flow rate is changed.

The experimental results show that for all the different configurations tested, the self-tuning controller is equal to or better than the PID compensator. The STC is found to be very robust in that it can handle deterministic disturbances (feed flow changes) as good as a well tuned PID compensator even though feedforward control action is not added. With the addition of feedfoward compensation, the STC performance is superior to that of the PID controller.

ACKNOWLEDGMENT

The author wishes to express his sincere gratitude to Dr. A. J. Morris and Dr. R. K. Wood for their assistance and guidance in this project. Special thanks are extended to Dr. Morris for providing the source programs used as the basis of the control algorithm implemented in this work.

Grateful acknowledgment is due to the staff (past and present) of the Data Acquisition Control and Simulation Centre in the Department of Chemical Engineering for their assistance in using their computing facilities.

Special thanks are due to all the staff in the instrument and mechanical shops for their assistance in keeping the pilot plant column in operation.

The author would like to acknowledge all his fellow graduate students for all their aid. A special note of thanks is in order for H. Kan for providing assistance in the actual experimental work.

Financial support given by the National Sciences and Engineering Research Council during this project is greatly appreciated.

Table of Contents

Chapter	Page
1. INTRODUCTION	1
1.1 INTRODUCTION	1
1.2 OBJECTIVE	2
1.3 THESIS ORGANIZATION	3
2. DEVELOPEMENT OF THE SELF-TUNING CONTROLLER	4
2.1 INTRODUCTION	4
2.2 LITERATURE SURVEY	6
2.2.1 Historical Development of Self-Tuning Regulator	6
2.2.2 Modification and Extension	8
2.2.3 Application	11
2.2.4 Convergence and Stability Analysis	14
2.3 DERIVATION OF THE SELF-TUNING CONTROLLER	14
2.3.1 Control Law Formulation	14
2.3.2 Closed Loop Response	23
2.3.3 Model Following For Set Point	26
2.3.4 Controller Parameter Estimation	26
2.3.5 Extension to Multivariable case	29
3. EXPERIMENTAL IMPLEMENTATION	33
3.1 DESCRIPTON OF EXPERIMENTAL EQUIPMENT	33
3.2 CONTROL LAW IMPLEMENTAION	37
4. SELECTION OF MANIPULATED VARIABLE	39
4.1 INTRODUCTION	39
4.2 ANALYSIS OF INTERNAL REFLUX RATIO SENSITIVITY TO HEAT DISTURBANCE	46

4.3 EXPERIMENTAL OPEN LOOP RESPONSES	51
4.4 CONCLUSION	57
5. EXPERIMENTAL RESULTS	59
5.1 INTRODUCTION	59
5.2 EXPERIMENTAL PROCEDURE	59
5.3 COMPUTER PROGRAMS	64
5.4 CONTROL BEHAVIOUR USING PID CONTROL	65
5.5 TOP COMPOSITION CONTROL	78
5.5.1 Control Performance Using the STC with Q-weighting	78
5.5.2 Control Performance Using the STC Without Q-weighting	86
5.5.3 Effect of Sampling Time	93
5.5.4 Effect on Control Performance when Identification is Stopped	96
5.6 BOTTOM COMPOSITION CONTROL WITH STC	99
5.7 SIMULTANEOUS CONTROL OF TERMINAL COMPOSITION	112
5.7.1 Multiloop Self-Tuning Control of Terminal Composition	112
5.7.2 Multivariable ST-Control of Terminal Composition	119
5.8 DISCUSSION OF RESULTS	122
6. CONCLUSIONS AND RECOMENDATIONS	127
NOMENCLATURE	133
REFERENCES	137
APPENDIX A: DETAIL SCHEMATIC OF PILOT SCALE DISTILLATION COLUMN	146
APPENDIX B: SAMPLE CALCULATION TO OBTAIN INITIAL Q-WEIGHTING PARAMETERS	148
APPENDIX C: PROGRAMS LISTING	150

LIST OF FIGURES

Figure	Page
CHAPTER TWO:	
2.1	Basic Structure of the Self-Tuning Regulator.....5
2.2	Block Diagram of Self-Tuning Controller for $G=EB$...24
2.3	Block Diagram of Self-Tuning Controller for $G = (EB+CQ)/Q$25
CHAPTER THREE:	
3.1	Schematic Diagram of the Distillation Column and Interface to Computers.....35
CHAPTER FOUR:	
4.1	Schematic Diagram of the Energy Balance Control Scheme.....41
4.2	Schematic Diagram of the Material Balance Control Scheme.....42
4.3	Open Loop Response to a 7 1/2% Step Decrease in Steam Flow With A Material Balance Configuration...53
4.4	Open Loop Response to a 7 1/2% Step Increase in Steam Flow With A Material Balance Configuration...54
4.5	Open Loop Response to a 7 1/2% Step Decrease in Steam Flow With an Energy Balance Configuration....55
4.6	Open Loop Response to a 7 1/2% Step Increase in Steam Flow With an Energy Balance Configuration....56

LIST OF FIGURES

Figure	Page
CHAPTER FIVE:	
5.1	PID Control of Top Composition for a +25% Step Change in Feed Flow Rate.....61
5.2	PID Control of Top Composition for a Step Decrease in Feed Flow Rate to its Normal Steady State Value.....62
5.3	PID Control of Top Composition for a -25% Step Change in Feed Flow Rate.....67
5.4	PID Control of Top Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value.....68
5.5	PID Control of Bottom Composition for a +25% Step Change in Feed Flow Rate.....69
5.6	PID Control of Bottom Composition for a Step Decrease in Feed Flow Rate to its Normal Steady State Value.....70
5.7	PID Control of Bottom Composition for a -25% Step Change in Feed Flow Rate.....71
5.8	PID Control of Bottom Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value.....72
5.9	Multiloop PID Control of Terminal Compositions for a +25% Step Change in Feed Flow Rate.....74

LIST OF FIGURES

Figure	Page
5.10 Multiloop PID Control of Terminal Compositions for a Step Decrease in Feed Flow Rate to its Normal Steady State Value.....	75
5.11 Multiloop PID Control of Terminal Compositions for a -25% Step Change in Feed Flow Rate.....	76
5.12 Multiloop PID Control of Terminal Compositions for a Step Increase in Feed Flow Rate to its Normal Steady State Value.....	77
5.13 Top Composition Behaviour When Operation of the STC With Q-WT is Initiated.....	80
5.14 Variation of G1 Parameters in the Initial Tuning Period for STC With Q-WT.....	81
5.15 Variation of F Parameters in the Initial Tuning Period for STC With Q-WT.....	82
5.16 ST Control of Top Composition for a -25% Step Decrease in Feed Flow Rate With Q-WT.....	84
5.17 ST Control of Top Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT.....	85
5.18 Top Composition Behaviour When Operation of the STC Without Q-WT is Initiated.....	87
5.19 Variation of G1 Parameters in the Initial Tuning Period for STC Without Q-WT.....	88
5.20 Variation of F Parameters in the Initial Tuning Period for STC Without Q-WT.....	89

LIST OF FIGURES

Figure		Page
5.21	ST Control of Top Composition for a -25% Step Decrease in Feed Flow Rate Without Q-WT.....	90
5.22	ST Control of Top Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value Without Q-WT.....	91
5.23	ST Control of Top Composition to a -25% Step Decrease in Feed Flow Rate Without Q-Wt: Two Minute Sampling Time.....	94
5.24	ST Control of Top Composition to a -25% Step Decrease in Feed Flow Rate Without Q-Wt: Three Minute Sampling Time.....	95
5.25	ST Control of Top Composition to a -25% Step Change in Feed Flow Rate Without Q-WT and the Identification Stopped.....	98
5.26	Top Composition Response After Resuming Identification in the STC.....	100
5.27	ST Control of Bottom Composition for a -25% Step Change in Feed Flow Rate With Q-WT ($k_p = 1.15$ and $k_i = 0.23$).....	102
5.28	ST Control of Bottom Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT ($k_p = 0.90$ and $k_i = 0.47$)...	103
5.29	ST Control of Bottom Composition for a +25% Step Change in Feed Flow Rate With Q-WT ($k_p = 0.90$ and $k_i = 0.47$).....	104

LIST OF FIGURES

Figure		Page
5.30	ST Control of Bottom Composition for a Step Decrease in Feed Flow Rate to its Normal Steady State Value With Q-WT ($k_p = 0.90$ and $k_i = 0.47$)...	105
5.31	STC + FF Control of Bottom Composition for a -25% Step Change in Feed Flow Rate With Q-WT ($k_p = 0.90$ and $k_i = 0.47$).....	108
5.32	STC + FF Control of Bottom Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT ($k_p = 0.90$ and $k_i = 0.47$).....	109
5.33	STC + FF Control of Bottom Composition for a +25% Step Change in Feed Flow Rate With Q-WT ($k_p = 0.90$ and $k_i = 0.47$).....	110
5.34	STC + FF Control of Bottom Composition for a Step Decrease in Feed Flow Rate to its Normal Steady State Value With Q-WT ($k_p = 0.90$ and $k_i = 0.47$).....	111
5.35	ML-ST Control of Terminal Compositions for a -25% Step Change in Feed Flow Rate With Q-WT (top: $k_p = 2.7$ $k_i = .45$ bot: $k_p = .90$ $k_i = .47$)...	115
5.36	ML-ST Control of Terminal Compositions for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT (top: $k_p = 2.7$ $k_i = .45$ bot: $k_p = .90$ $k_i = .47$).....	116

LIST OF FIGURES

Figure		Page
5.37	ML-STC + FF (bottom) Control of Terminal Compositions for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT (top: $k_p - 2.7 k_i - .45$ bot: $k_p - .90 k_i - .47$)...	117
5.38	ML-STC + FF (bottom) Control of Terminal Compositions for a +25% Step Change in Feed Flow Rate With Q-WT (top: $k_p - 2.7 k_i - .45$ bot: $k_p - .90 k_i - .47$).....	118
5.39	MV-ST Control of Terminal Compositions for a -25% Step Change in Feed Flow Rate With Q-WT (top: $k_p - 1.5 k_i - .23$ bot: $k_p - .90 k_i - .47$)...	120
5.40	MV-ST Control of Terminal Compositions for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT (top: $k_p - 1.5 k_i - .23$ bot: $k_p - .90 k_i - .47$).....	121
5.41	Variation of G1 Parameters in the Transition Period of a -25% Step Change in Feed Flow Rate for the Bottom STC with Q-WT.....	124
5.42	Variation of F Parameters in the Transition Period of a -25% Step Change in Feed Flow Rate for the Bottom STC with Q-WT.....	125

LIST OF TABLES

Table		Page
4.1	Variations of V/L With Control Scheme for $\pm 20\%$ Changes in Vapor Rate at an External Reflux Ratio of 10	48
4.2	Variations of V/L With Control Scheme for $\pm 20\%$ Changes in Vapor Rate at an External Reflux Ratio of 1.0	49
4.3	Variations of V/L With Control Scheme for $\pm 20\%$ Changes in Vapor Rate	50
4.4	Top and Bottom Composition Changes to Steam Flow Rate Disturbances	52
5.1	Summary of Absolute Error Values for PID Control ..	73
5.2	Controller Parameters and Options for STC Top Composition Control	79
5.3	Summary of Top Composition Absolute Error Values (Over 60 Samples) for STC With Q-Weighting	83
5.4	Comparison of Top Composition Absolute Error Values (Over 60 Samples) for STC With and Without Q-Weighting	86
5.5	Comparison of Top Composition Absolute Error Values Calculated for a Period of One Hour After the Feed Flow Rate Disturbance for Different Sampling Time	96
5.6	Controller Parameters and Options for STC Bottom Composition Control	101

LIST OF TABLES

Table	Page
5.7	Comparison of Bottom Composition Absolute Error Values (over 1 1/2 hour) for PID and STC106
5.8	Comparison of Bottom Composition Absolute Error Values (over 1 1/2 hour) for PID, STC and STC with Feedforward107
5.9	Controller Parameters and Options for Multiloop and Multivariable STC113
5.10	Summary of Top and Bottom Compositions Absolute Error Values for Multiloop STC With and Without Feedforward (FF) added114
5.11	Comparison of Top and Bottom Composition Absolute Error Values for Multiloop PID, Multiloop STC (with and without FF) and Multivariable STC122

1. INTRODUCTION

1.1 INTRODUCTION

Interest in improving terminal composition control strategies for distillation columns has increased in view of the dramatic increases in energy cost. With better control, less over purification is needed to guarantee product specification. The result is reduced expenditure in both heating and cooling medias hence lowering energy requirements.

Over the years, considerable effort has been made to find better control methods that would improve the control performance over that of the conventional feedback proportional-integral- derivative (PID) controller [1]. Some of the approaches have been implemented on the pilot plant distillation column at the University of Alberta with varying degree of success [2-9]. The difficulties experienced with many of the strategies that have been tested stem from the fact that the techniques were designed to handle linear systems. Unfortunately, the distillation column is clearly a nonlinear system [9]. When a controller is tuned to perform well for one disturbance, the control behavior for another disturbance, or set point change may not be satisfactory. To compensate for nonlinearity, the controllers have to be retuned. An ideal controller would be an adaptive controller where the parameters of the controller change to compensate for changes in process

condition. The self-tuning regulator (STR) is such a controller [10]. In the self-tuning regulator, a recursive identification scheme is used to identify either the parameters of a stochastic model for the process or the controller parameters directly. In either case, the control parameters will be changed to suit changes in operating condition.

The self-tuning regulator has been developed from work done by Åström and Wittenmark [10-11], Peterka [12] and Clarke et al. [13-18]. As originally proposed, the controller was for a single input-single output (SISO) system. Borisson [19], Kevinsky et al. [20-21] and Morris et al. [22-24] extended the work to handle a class of multivariable cases.

1.2 OBJECTIVE

The object of this study is to experimentally evaluate the performance of the self-tuning controller (STC) as modified by Morris et al. [24] on the pilot plant distillation column at the University of Alberta. The STC will be studied in single loop, multiloop and multivariable configuration. Results for the self-tuning controller will be compared to those for well tuned proportional-integral-derivative (PID) controllers. Both the STC and PID controller will be used to control the terminal composition of the distillation column when the column is subjected to feed flow disturbances.

1.3 THESIS ORGANIZATION

The thesis will be separated into six chapters. The first chapter gives an introduction. Chapter two contains a survey of the work done to date and a brief discussion of the theory underlying the STC. Extensions to the originally proposed scheme by Åström and Wittenmark [10] will be presented and the actual algorithms used in this study will be derived. Chapter three describes the experimental equipment used in this study. Chapter four deals with the selection of the manipulated variable for control of distillate composition. An analysis of the internal reflux ratio will be given to show which of the manipulated variables, reflux flow or distillate flow should be used. Experimental open loop responses will be presented to show the sensitivity of distillate composition to reflux flow and distillate flow manipulations. Chapter five contains the experimental results. The results will be presented in the following manner:

- a. single loop control for overhead composition;
- b. single loop control for bottom composition;
- c. simultaneous control in multiloop configuration;
- d. simultaneous control in multivariable configuration.

Finally, Chapter six summarizes the conclusions and recommendations borne out by this study.

2. DEVELOPEMENT OF THE SELF-TUNING CONTROLLER

2.1 INTRODUCTION

The self-tuning regulator as proposed by Åström [10] as the name implies is an adaptive controller. The algorithm consists of two stages at every sample interval. The first stage is a recursive least square identification scheme to obtain the parameters of a linear stochastic model used to describe the process being controlled while in the second stage a control signal is calculated using minimum variance of the output as a criterion. The parameters estimated from the identification step are used directly as if they are the true parameters. Åström [10] has shown that for systems with constant parameters, if the parameters converge, the controller will converge to the true minimum variance controller. The structure of the self-tuning regulator is shown in Figure 2.1.

There are many advantages in using a self-tuning regulator. First, it is an adaptive controller. Since the parameters of the model are being constantly updated, changes in the process parameters are automatically accounted for. For a nonlinear plant, the parameters identified will give a linearized approximate model of the nonlinear plant. As the operating conditions change, the parameters readjust to linearize the model around the new operating point. For a plant with time varying properties, again the parameters will adapt to the new condition.

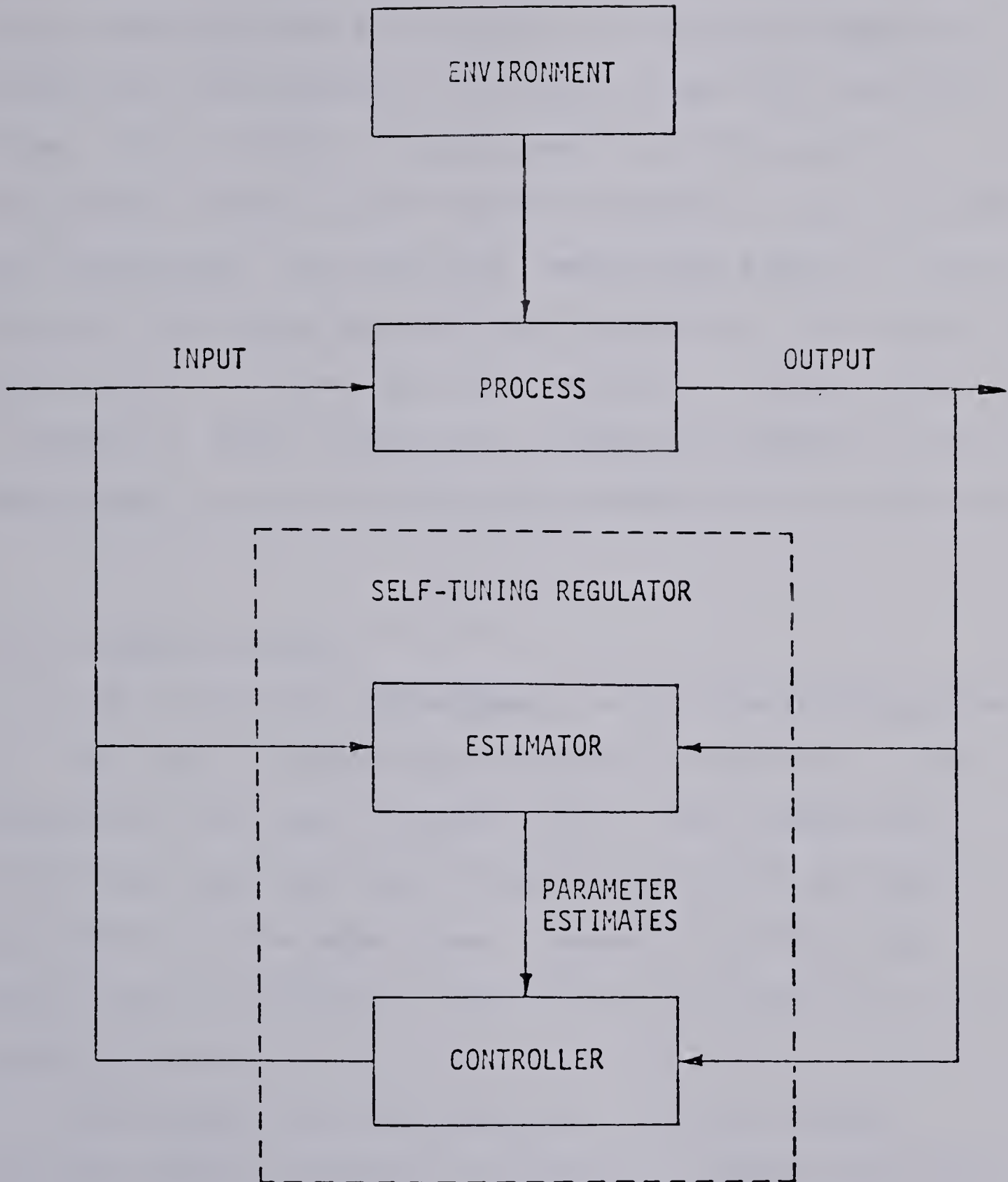


Figure 2.1 Basic Structure of the Self-Tuning Regulator

Second, as it is self-tuning, limited knowledge of the system to be controlled is needed. There is no need to do a lengthy background identification test to obtain model parameters. Although certain parameters have to be specified, these parameters are readily obtained in comparison to determining the constants for a well tuned PID controller. Three, since a model is being used to do the control calculation, there is an inherent predictor. For a system with time delay, the predictor removes the effect of time delay on the system characteristic equation. The process is controlled as if there was no time delay. Finally, by using a stochastic model formulation, random disturbances such as measurement or process noise are automatically accommodated.

2.2 LITERATURE SURVEY

2.2.1 Historical Development of Self-Tuning Regulator

The idea of combining estimation and control in one algorithm is not new. Kalman [25] in 1958 formulated a simplified algorithm that attempted to do both estimation and control at the same time. However, the theory was insufficient and digital computers were not available for process control.

Åström was first involved with the self-tuning regulator when he and Bohlin presented a general form for stochastic models [26]. They considered the maximum likelihood technique for identifying the parameters of linear stochastic models structured according to their

general form. Subsequently, Åström [27] looked at the accuracy of the estimates obtained and the possibility of using recursive least squares to identify the parameters.

Peterka [12] then used the Åström-Bohlin model formulation to derive an algorithm that combined estimation of the model parameters and calculation of the control law. He used recursive least squares to identify the parameters of the Åström-Bohlin model; with the model estimates available, a minimum variance of the output was used as the criterion for determining controller action. For some conditions, Peterka found that the controller used was identical to the minimum variance controller that could be derived had the model parameters been known.

Following Peterka's work, Åström and Wittenmark [28] did a thorough study looking at separate and combined estimation and control strategies. They considered models that have constant unknown parameters and models that are time-varying Gauss-Markov processes. With Peterka's results and their own work, Åström and Wittenmark formulated the self-tuning regulator [10]. They proved two theorems concerning the self-tuning regulator. Using weak assumptions, they showed that if the parameters converged, then the covariance of the output variable and the crosscovariance between the output and input variable would tend to zero. Furthermore, it was shown for systems with constant parameters that if the parameters converged, the controller would converge to the minimum variance controller

that could be derived if the model parameters were known. Results of a detailed simulation study using the self-tuning regulator were presented by Wittenmark [29].

A more complete survey of the different studies that led to the formulation of the self-tuning regulator has been given by Chang [30]. The papers by Clarke [18] and Åström [11] are excellent reviews of many of the self-tuning studies that have been performed.

2.2.2 Modification and Extension

In its original formulation, the self-tuning regulator had very few options. For instance, there was no set point following capability. Many extensions and modifications have since been added to increase the generality and robustness of the controller.

Wittenmark has extended the regulator by adding features to allow set point following for the output variable and feedforward action for measurable disturbances [29].

Clarke et al. [13-18] have presented a generalized version of the self-tuning regulator. The term self-tuning controller designation is used for the Clarke approach to distinguish this type of controller from the self-tuning regulator for the Åström approach. In Clarke's scheme, instead of identifying the parameters of a model and then subjecting the identified model to some control criterion, the controller parameters are identified directly. In addition, the cost function for the controller is modified

to include control effort, process output and set point following. Inclusion of control effort into the performance criteria achieves two objectives. Adding a penalty for control effort causes the control output to remain within the constraint limits. The 'bang-bang' controller response that is observed with minimum variance controller is prevented. Second, it can be shown that with the appropriate choice of control weighting, systems that are nonminimum phase can be made closed-loop stable [14]. In a treatment of nonminimum phase systems, Cegrell and Hedqvist [31] have proposed a similar solution to that used by Clarke. Åström [32], Åström and Wittenmark [33], Turtle and Phillipson [34] and Peterka [12] have suggested the use of suboptimal strategies to handle such systems.

Morris [22-24] in turn has generalized the work presented by Clarke. He has introduced a discrete compensator in PID form for the control weighting used in the cost function. The PID controller is suggested because no offset results when a set point change is made and furthermore, the characteristics of the PID algorithm are well understood. Set point model following is added to allow the process output to change according to any user specified response when a set point change is made; e.g., the output can be made to follow a first order response with some specified time constant. By allowing the output to change slowly, there is no need for excessive control action at the start of a set point change.

Morris has further extended the controller to handle multivariable systems with an equal number of inputs and outputs. The multivariable system is reduced to a series of decoupled single loop controllers. The decoupling is achieved by using the feedforward option to handle the interaction between the individual loops; i.e., the interactions between the loops are treated as measurable disturbances. This approach removes the need to identify directly the parameters (matrices) of a multivariable system. Direct identification of such system has been considered by Gustavsson et al. [35].

Åström and Peterka have considered extending the self-tuning regulator to the multivariable case for systems with constant and unknown parameters [36]. They looked at identifying the parameters of the multivariable model directly. Borison [19] and Keviczky [20] have presented algorithms to handle multivariable systems that identify the parameters of the multivariable structure directly. De Keyser [37] has proposed a multivariable self-tuning scheme that calculates a control output based on the identified estimates and the covariance of the estimates. With this approach, excessive control is prevented when the parameters are bad; in effect, a more cautious controller is devised.

Different identification schemes have been considered for the self-tuning regulator. Aside from normal least squares, other methods that have been tried include: stochastic approximation [38], extended least squares [39]

and maximum likelihood [40]. Recently, it has been shown that the standard least square identification method can lead to problems when a system experiences no disturbances over a long period of time [24]. Without any disturbance, the identification scheme receives no new information and as a result, the covariance matrix tends to 'blow-up' or lose positive definiteness. In either case, the parameters will diverge causing stability problems with the controller. To overcome this problem, Morris [24] has proposed modifying the normal least square scheme by simply updating the square root of the covariance matrix. Morris also proposed to employ a recursive learning estimator [24] once reasonable estimates have been obtained with normal least squares. Clarke [18] has reported alternate solutions which modify the identification after a 'no disturbance' situation is detected. One possibility would be to stop the identification if the determinant of the covariance matrix exceeds some specified limit or to introduce a disturbance if this condition occurs. The third approach Clarke suggested was that the forgetting factor be adjusted so that it tends to unity if no disturbance is entering the system.

2.2.3 Application

Numerous applications of the self-tuning regulator for the control of various different types of systems have been reported. Those considered pertinent to this work will be briefly discussed. Wittenmark and Borisson [40] have used

the self-tuning regulator for controlling the moisture content of a paper machine. They found that the regulator was able to handle nonstationary disturbances and that the choice for the number of parameters in the model was unimportant. Cegrell and Hedqvist, on a different paper machine, used a self-tuning regulator with five inputs and one output [41]. They reported that the self-tuning regulator, in comparison to a PI regulator, reduced the output variance during quality change and mill step-up. In addition, their experience was consistent with that of Borisson and Wittenmark in that they also concluded that the number of parameters in the model was unimportant.

Borisson and Syding have reported the use of a self-tuning regulator on an ore-crusher [42]. They found that the regulator was able to adapt to changes in the characteristics of the ore feed and the crusher itself. As a direct result of using the self-tuner, a 10% increase in production was achieved over that possible using PI control.

Dumont and Belanger [43-44] have used the self-tuning regulator for controlling a titanium oxide kiln. They were able to reduce the number of parameters required in the self-tuning regulator by applying the regulator on a plant that was already controlled by a fixed parameter regulator. They reported that with this type of strategy, instead of a normal 10-12 hours of offspecification product during a grade change, the switchover time was reduced to 2 hours.

Keviczky et al. [21] and Westerlund et al. [45] have

applied the self-tuning regulator to control of cement raw material mixing. The application by Keviczky was on a multivariable system and the control criteria was based on minimum variance of the output and some required average over a time duration. In both applications, it was found that the STR gave results that were superior to those achieved with a conventional control approach.

Various authors have presented results from applications of the self-tuning regulator to pilot scale equipment. Sastry et al. [46] reported on the performance of the self-tuning regulator for controlling the overhead composition of a binary distillation column. In general, the self-tuning regulator performed better than a PI controller. Chang [30] used the self-tuner on a pilot scale evaporator. He considered how the various design parameters affected the performance of the regulator. Harris et al. [47] studied the control of a pilot scale packed bed reactor with the self-tuning controller. The reaction involved was extremely temperature sensitive which resulted in highly nonlinear and nonminimum phase system behaviour. The self-tuning controller was reported to be superior to the conventional PI algorithm. The study also looked at the problem of 'parameter-windup'; this is a situation when the parameters starts to diverge because the controller output is clamped at one of the constraints.

Applications of the self-tuning regulator have also been reported for the control of an enthalpy exchanger[48],

digester[49], absorption column[50] and ship steering[51].

2.2.4 Convergence and Stability Analysis

Ljung has presented results for convergence analysis of general recursive stochastic algorithms [52-56]. His approach was to associate the algorithms with an ordinary differential equation so that the convergence properties of the algorithm could be obtained by looking at the stability properties of the differential equation. Åström has used Ljung's results to analyze the self-tuning regulator [11].

For the self-tuning controller, convergence analysis is based on the idea of treating the controller as a feedback system [57,58]. Using the Martingale theorem [59,60], Gawthrop [61] has shown that under certain conditions, convergence will occur with a probability of 1.0.

2.3 DERIVATION OF THE SELF-TUNING CONTROLLER

2.3.1 Control Law Formulation

A single input-single output (SISO) stochastic process can be described by:

$$Ay_t = z^{-k}Bu_t + C\xi_t + z^{-k^L}Lv_t \quad (2.1)$$

where A, B, C, L are polynomials in z^{-1} with $a_0 = 1$, $b_0 = 0$ and $c_0 = 1$ with $z^{-N}y_t$ denoting the value of y , N samples in the past. The terms y_t , u_t , ξ_t , v_t are respectively the process output, control input, random disturbance and measurable disturbance. The system is considered to be of order n with

a time delay equal to k times the sample time and k^L the time delay associated with the known disturbance. It is also assumed that C has roots that are within the Z -domain unit circle and that ξ_t is a random uncorrelated noise signal with zero mean. The controller is designed to minimize the cost function:

$$J_1 = E\{(Py_{t+k} - Rw_t)^2 + (Q'u_t)^2\} \quad (2.2)$$

where P, R, Q' are polynomials in z^{-1} which allow weighting to be applied to the output, set point and control effort. The term $(Py_{t+k} - Rw_t)$ provides set point following while $(Q'u_t)$ puts a penalty on control action. From the form of the cost functional it can be seen that if $Q'u_t$ does not tend to zero at steady state, $Py_{t+k} - Rw_t$ will tend to some constant other than zero and the process output will exhibit an offset (cf. equation (2.23)).

In order to minimize J_1 , an expression is required for y_{t+k} in terms of past input and output data. Here, y_{t+k} denotes the k -step ahead predictor for the output. The basic derivation of the self-tuning regulator was given first by Åström [10]. However, the ensuing derivation follows closely the formulation given by Clarke [13] and Morris and coworkers [24].

Expressing Equation (2.1) in terms of the k -step ahead predictor gives:

$$y_{t+k} = \frac{B}{A} u_t + z^k \frac{C}{A} \xi_t + z^{k-k^L} \frac{1}{A} v_t \quad (2.3)$$

In Equation (2.3), at time t , providing k^L is larger than k , only the future random disturbances from time $t+1$ to $t+k$ are unknown. Separating $z^k(C/A)\xi_t$ into disturbances up to the present time and those at future time gives:

$$z^k \frac{C}{A} \xi_t = E' z^k \xi_t + \frac{F'}{A} \xi_t \quad (2.4)$$

where $E' z^k \xi_t$ represents the future disturbances and $(F'/A)\xi_t$ equals the disturbances up to the present time. The polynomials E' and F' have terms:

$$E' = 1 + e'_1 z + \dots + e'_{k-1} z^{-(k-1)} \quad (2.5)$$

$$F' = f'_0 + f'_1 z + \dots + f'_{n-1} z^{-(n-1)} \quad (2.6)$$

Substitution of Equation (2.4) into the predictor equation (2.3) and noting that $z^k \xi_t$ is the same as gives:

$$y_{t+k} = \frac{B}{A} u_t + E' \xi_{t+k} + \frac{F'}{A} \xi_t + z^{k-k^L} \frac{L}{A} v_t \quad (2.7)$$

The random disturbance ξ_t up to time t can be expressed in terms of previous inputs and outputs by rewriting Equation (2.1) in terms of ξ_t as:

$$\xi_t = \frac{A}{C} y_t - z^{-k} \frac{B}{C} u_t - z^{-k^L} \frac{L}{C} v_t \quad (2.8)$$

Using Equation (2.8) to substitute for ξ_t in Equation (2.7), after some rearrangement yields:

$$y_{t+k} = E' \xi_{t+k} + \left[\frac{1}{A} - \frac{z^{-k} F'}{AC} \right] B u_t + \frac{F'}{C} y_t + \left[\frac{1}{A} - \frac{z^{-k} F'}{AC} \right] z^{k-k^L} L v_t \quad (2.9)$$

From Equation (2.4), it follows that the terms in the brackets simplify to E'/C so Equation (2.9) becomes:

$$y_{t+k} = E' \xi_{t+k} + \frac{E' B}{C} u_t + \frac{F'}{C} y_t + \frac{E' L}{C} z^{k-k^L} v_t \quad (2.10)$$

Since ξ_t has been assumed to be a white noise sequence, the future disturbance is expected to be zero. Taking the expectation of Equation (2.10), the predicted output is given as:

$$y_{t+k}^* = \frac{E' B}{C} u_t + \frac{F'}{C} y_t + \frac{E' L}{C} z^{k-k^L} v_t \quad (2.11)$$

where the asterisk denotes a predicted quantity.

All the terms on the right hand side of Equation (2.11) are known at time t provided k^L is equal to or greater than k ; i.e., the time delay in the feedforward path must be larger than the time delay for the control action. If the above condition is not true, a predictor for future measured disturbances is needed. However, for most actual processes, the time delay for the measured disturbance is much larger than the time delay for the control variable.

The performance criteria given by Equation (2.2) requires a weighted output. Applying 'P' weight to Equation (2.3) yields:

$$Py_{t+k} = \frac{PB}{A} u_t + \frac{PC}{A} \xi_{t+k} + z^{k-k^L} \frac{PL}{A} v_t \quad (2.12)$$

Now by defining:

$$P = P_N / P_D \quad (2.13)$$

and,

$$\frac{PC}{A} = E + \frac{z^{-k} F}{AP_D} \quad (2.14)$$

Using Equations (2.12) to (2.14) and following the derivation used for the k-step predictor (Equation (2.10)), the weighted output Py_{t+k} can be written as:

$$Py_{t+k} = E\xi_{t+k} + \frac{F}{CP} y_t + \frac{EB}{C} u_t + \frac{EL}{C} z^{k-k^L} v_t \quad (2.15)$$

As before, the best estimate of future disturbances is zero so the k-step ahead weighted output predictor is given by:

$$(Py_{t+k})^* = \frac{F}{CP} y_t + \frac{EB}{C} u_t + \frac{EL}{C} z^{k-k^L} v_t \quad (2.16)$$

The actual weighted output and the predicted output are related by:

$$Py_{t+k} = (Py_{t+k})^* + \epsilon_{t+k} \quad (2.17)$$

where, $\epsilon_{t+k} = E\xi_{t+k}$. Substituting the weighted output given by Equation (2.17) into the cost function yields:

$$J_1 = E\{ [(Py_{t+k})^* + \epsilon_{t+k} - Rw_t]^2 + [Q'u_t]^2 \} \quad (2.18)$$

Since it has been assumed that the random disturbances are uncorrelated with either the input or output, the expectation simplifies to:

$$J_1 = E\{ [(Py_{t+k})^* - Rw_t]^2 \} + E\{ [Q'u_t]^2 \} + \sigma_{t+k}^2 \quad (2.19)$$

where σ_{t+k}^2 is the variance of the random disturbances.

The cost function is minimized by setting the partial derivative $\partial J_1 / \partial u_t$ to zero. To obtain the true optimal solution, the expectation operator must be included in the differentiation [62]. However, a complicated control law results so instead, a stage by stage optimal controller is used. In this work, following the approach of Clarke and Gawthrop [14], the expectation is removed from Equation (2.19) and the resulting equation is differentiated with respect to u_t . The controller that results is optimal each time the control calculation is performed. Although the solution is not globally optimal, Harris [47] has shown that the control response of the suboptimal strategy is close to the result that is obtained when the true optimal controller is used. Removing the expectation operator, the partial derivative becomes:

$$\frac{\partial J_1}{\partial u_t} = 2[(Py_{t+k})^* - Rw_t] * \frac{\partial (Py_{t+k})^*}{\partial u_t} + 2Q'u_t * \frac{\partial (Q'u_t)}{\partial u_t} \quad (2.20)$$

Performing the necessary differentiation and noting that $(Py_{t+k})^*$ is related to u_t (cf. Equation (2.16)), setting Equation (2.20) to zero gives:

$$\left[(Py_{t+k})^* - Rw_t \right] * \frac{e_0 b_0}{c_0} + q_0' Q' u_t = 0 \quad (2.21)$$

where e_0 , b_0 , c_0 and q_0' are the leading coefficients for the polynomials E , B , C and Q' . Defining:

$$Q = \begin{bmatrix} q_0' \\ -b_0 \end{bmatrix} * Q' \quad (2.22)$$

and recalling that e_0 and c_0 are equal to 1, Equation (2.21) can be written as:

$$u_t = \frac{1}{Q} \left[Rw_t - (Py_{t+k})^* \right] \quad (2.23)$$

Examination of Equation (2.23) shows that it is similar to a standard feedback control law with the main difference that the feedback value is a predicted output instead of the actual measurement. It follows that the inverse of the control effort weighting is the discrete compensator used in the feedback loop. While any discrete compensator can be used, the PI compensator is selected because its characteristics are widely known. The PI compensator used is of the form:

$$G = \frac{1}{Q} = \frac{k_p (1 - z^{-1}) + k_i}{(1 - z^{-1})} \quad (2.24)$$

The PI compensator is used instead of a PID compensator because derivative action is generally not required for systems without time delay. As the self-tuning controller

predicts the output, the effect of process time delay is removed hence derivative action is not necessary. The incremental PI form is used to insure that Qu_t tends to zero at steady state since this is the requirement for zero offset. If Qu_t does not tend to zero at steady state, the difference between the weighted setpoint and the predicted output ($Rw_t - (Py_{t+k})^*$) is different from zero (cf. Equation (2.23)) At steady state, the expected predicted output is equal to the actual output. If Qu_t does not equal zero, the difference between the weighted set point and output is different from zero so an offset occurs.

The control law can now be formulated in terms of a predicted scalar output defined as:

$$\emptyset_{t+k}^* = (Py_{t+k})^* - Rw_t + Qu_t \quad (2.25)$$

The control law formulation thus involves the calculation of u_t by zeroing \emptyset_{t+k}^* at every sample instant. The actual scalar output is given by:

$$\emptyset_{t+k} = Py_{t+k} - Rw_t + Qu_t \quad (2.26)$$

As ϵ_{t+k} and Py_{t+k} are uncorrelated, the error in the estimates of \emptyset_{t+k} can be obtained from Equation (2.17) as:

$$\emptyset_{t+k}^* = \emptyset_{t+k} - \epsilon_{t+k} \quad (2.27)$$

Substitution of the expression for $(Py_{t+k})^*$ from Equation (2.16) into Equation (2.25) yields:

$$C\emptyset_{t+k}^* = F/P_d y_t + Ebu_t + CQu_t - CRw_t + ELz^{k-k^L} v_t \quad (2.28)$$

Defining:

$$\bar{y}_t = y_t / P_d \quad (2.29)$$

$$\bar{w}_t = R w_t \quad (2.30)$$

$$\bar{v}_t = z^{k-k^L} v_t \quad (2.31)$$

$$H = -C \quad (2.32)$$

$$D = EL \quad (2.33)$$

allows the control law to be written as:

$$C\emptyset_{t+k}^* = F\bar{y}_t + E B u_t + C Q u_t + H \bar{w}_t + D \bar{v}_t \quad (2.34)$$

Furthermore, the polynomial associated with the control output u_t can be defined to be independent of the control weight function by setting:

$$G = EB \quad (2.35)$$

which then allows the control law to be expressed as:

$$C\emptyset_{t+k}^* = F\bar{y}_t + G u_t - H Q u_t + H \bar{w}_t + D \bar{v}_t \quad (2.36)$$

Alternatively, all the polynomials involving u_t can be grouped together in the form of:

$$G = \frac{EB + QC}{Q} \quad (2.37)$$

So the control law takes the form:

$$C\emptyset_{t+k}^* = F\bar{y}_t + G Q u_t + H \bar{w}_t + D \bar{v}_t \quad (2.38)$$

The block diagram of the control law as expressed in Equations (2.36) and (2.38) are shown in Figures 2.2 and 2.3 respectively.

2.3.2 Closed Loop Response

The closed loop response can be obtained by substituting $(Py_{t+k})^*$ given by Equation (2.17) into the control law Equation (2.23). After rearranging the output, y_t is given by:

$$y_t = \frac{1}{p} \{E\xi_t + z^{-k}Rw_t - z^{-k}Qu_t\} \quad (2.39)$$

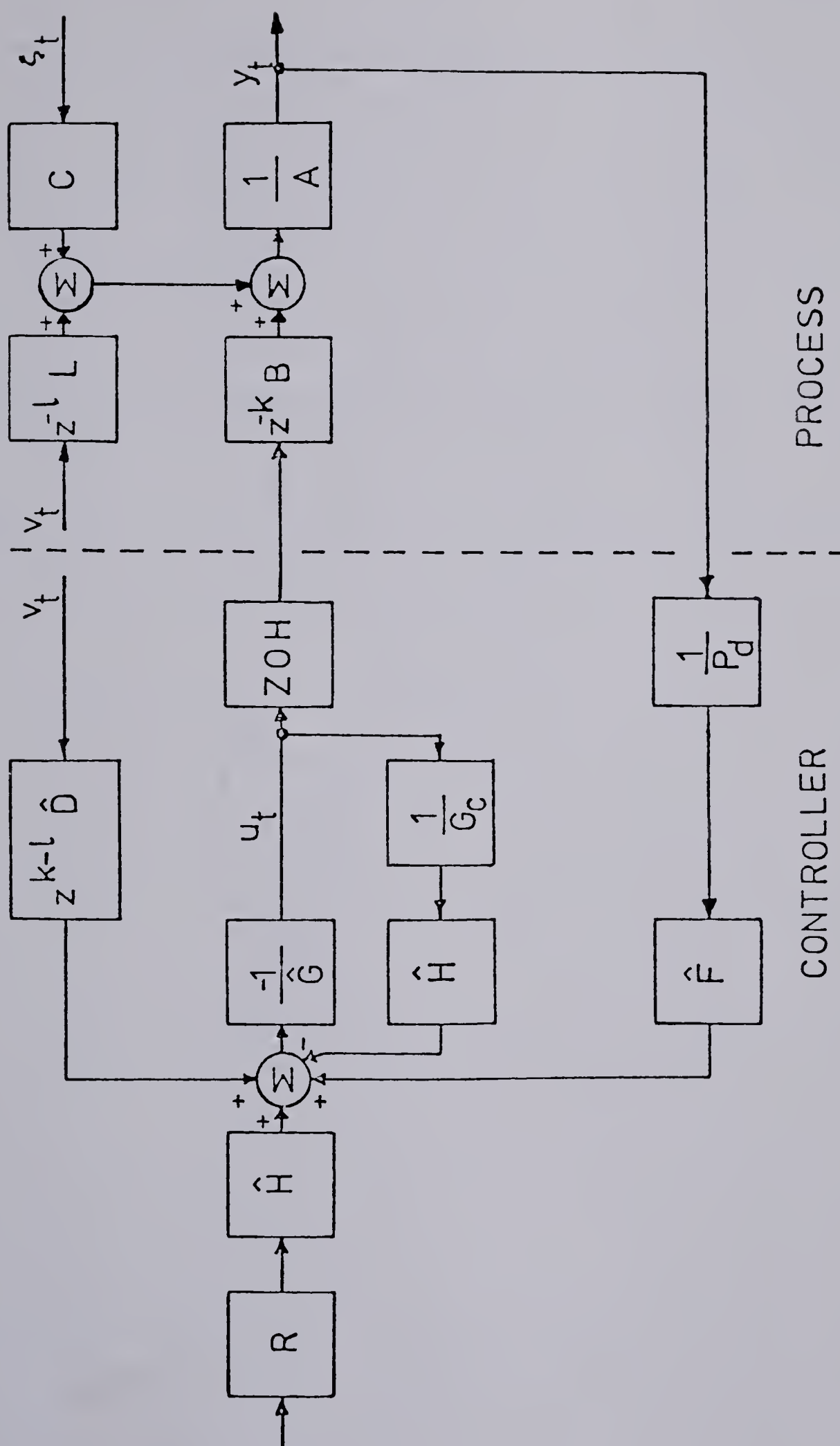
Solving Equation (2.1) for $z^{-k}u_t$ and substituting into Equation (2.39) yields the closed loop response:

$$y_t = \frac{1}{p} \left[\frac{(EB + CQ)\xi_t + z^{-k}BRw_t + z^{-kL}QLv_t}{(B + QA/P)} \right] \quad (2.40)$$

The stability of the system is governed by the characteristic equation:

$$PB + QA = 0 \quad (2.41)$$

It is clear that if B is nonminimum phase; i.e. its roots lie outside the z -domain unit circle, by proper choice of Q , the roots of the characteristic equation will still be inside the unit circle. The closed loop response is then stable. It is to be noted that as in the case of the analytical predictor [63] and Smith predictor [64], the effect of the system time delay (z^{-k}) has been removed from

Figure 2.2 Block Diagram of Self-Tuning Controller for $G=EB$

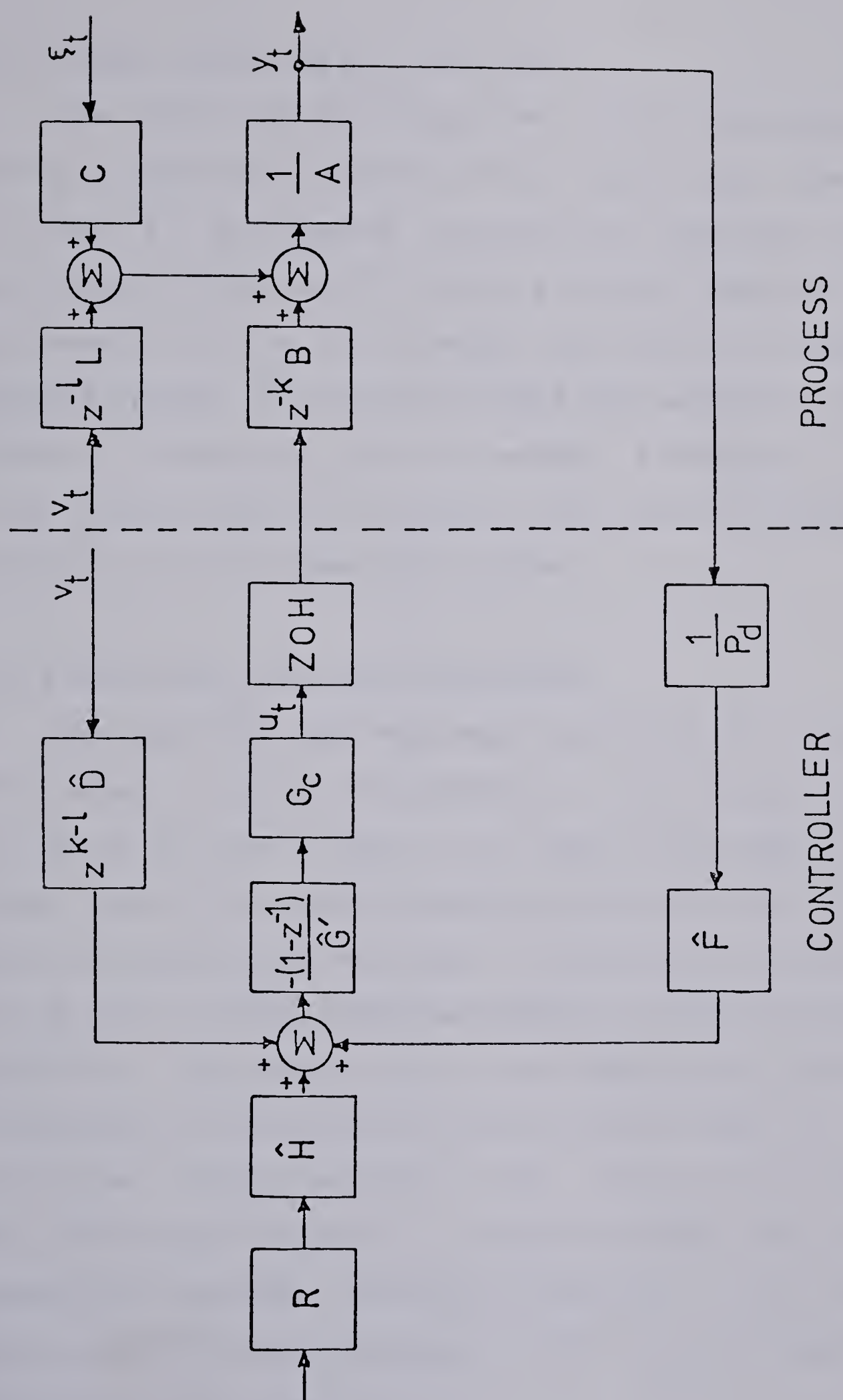


Figure 2.3 Block Diagram of Self-Tuning Controller for $G=(EB+CQ)/Q$

the characteristic equation.

2.3.3 Model Following For Set Point

The ability of the algorithm to force the output to follow a specified response can be clearly seen from Figures 2.2 and 2.3. By changing the weighting functions P and R , the output can be made to follow a desired response. Furthermore, the set point weighting function R can be changed without altering the closed loop stability of the system. In contrast, if the P weight is changed, it will alter the characteristic equation and possibly change the stability of the closed loop system.

2.3.4 Controller Parameter Estimation

The algorithm that has been derived is for a system with known structure and parameters. Yet in practice, this is rarely the case. Even if the order of the model is known, usually the exact process dynamics and hence the model parameters are not known. One possible solution is to use an on-line identification scheme to obtain parameter estimates. The control law can be rendered less sensitive to parameter bias and variation by using a specific structured 'controller-model' [24]. This is true provided the performance function J_1 is minimized when the estimated parameters are used. Subtracting Equation (2.34) from Equation (2.27) and rearranging yields ϕ_{t+k} in linear regression form, that is:

$$\emptyset_{t+k} = \underline{X}_t^T \underline{\theta}_t + (1-C)\emptyset_{t+k}^* + \varepsilon_{t+k} \quad (2.42)$$

where:

$$\underline{\theta}_t = [F_t, G_t, D_t, H_t]$$

$$\underline{X}_t = [\bar{y}_t, u_t, \bar{v}_t, \bar{w}_t]$$

The parameters in the $\underline{\theta}_t$ vector can now be estimated using a recursive least squares algorithm [65]. At every sample interval, the parameter estimate vector $\hat{\underline{\theta}}_t$ is updated according to:

$$\emptyset_t = P\bar{y}_t - R\bar{w}_{t-k} + Qu_{t-k} \quad (2.43)$$

$$\hat{\underline{\theta}}_t = \hat{\underline{\theta}}_{t-1} + \underline{K}_t (\emptyset_t - \underline{X}_{t-k}^T \hat{\underline{\theta}}_{t-1}) \quad (2.44)$$

$$\underline{K}_t = \frac{\underline{P}^C_t \underline{X}_{t-k}}{(\rho + \underline{X}_{t-k}^T \underline{P}^C_t \underline{X}_{t-k})} \quad (2.45)$$

$$\underline{P}^C_{t+1} = \frac{1}{\rho} \underline{P}^C_t (1 - \underline{K}_t^T \underline{X}_{t-k}) \quad (2.46)$$

Here, \underline{K}_t is the Kalman gain vector and \underline{P}^C is a symmetric matrix that is proportional to the covariance matrix. To allow tracking of slow time varying parameters, a forgetting factor ρ is used to weight out the effect of past data or alternatively, a positive definite matrix can be added to the \underline{P}^C update relationship to weight out old data. In both cases, when the system experiences no disturbance over a long period, the \underline{P}^C matrix can become

extremely large [24]. This occurs because without any disturbances, no new information can be obtained by the identification algorithm. As a result, the Kalman gains become small and from Equation (2.46), $\underline{\underline{p}}^C(t)$ approaches the value $\underline{\underline{p}}^C(t-1)/\rho$. If the system remains undisturbed over a period of time t , then $\underline{\underline{p}}^C(t)$ is equal to $\underline{\underline{p}}^C(t_0)/\rho^t$, where t_0 refers to the time when the system last experienced a disturbance. As the forgetting factor (ρ) is less than one, as time passes, $1/\rho^t$ becomes very large hence causing the matrix to 'blow-up'. When the $\underline{\underline{p}}^C$ matrix becomes large, limited machine precision may cause the matrix to lose positive definiteness. As a result, the parameters will diverge causing instability in the control loop. The proposed solution, and that adopted in this study, is to modify the identification scheme to update the square root of the $\underline{\underline{p}}^C$ matrix [24]. The square root will always be positive definite and it is inherently of higher numerical precision. Consequently, the control algorithm now consists of using a recursive square root identification to obtain the parameter vector $\hat{\underline{\theta}}_t$ and then using the estimates to calculate a new control output according to the criteria (cf. equation (2.38)):

$$\underline{\underline{x}}_t^T \hat{\underline{\theta}}_t = 0 \quad (2.47)$$

Åström has shown that by using a control law of the form given by Equation (2.47), all vectors with the form $\underline{\theta} = \alpha \hat{\underline{\theta}}_t$ will produce the same control [11]. This means that

the parameters will not necessarily converge to provide a unique solution. If the parameters converge to an α that is too large or small, numerical problems could result. In addition, as the estimates are being used in the feedback controller, they are correlated with the disturbance. It may then not be possible to identify all the parameters in the $\underline{\theta}$ vector. Åström has proposed that one parameter be fixed to overcome these problems. One obvious choice is to fix the leading coefficient for the set point weighting polynomial providing w_t is not zero. In this case, h_0 should be set to -1 as it is equal to $-c_0$. Alternatively, Åström has suggested fixing the leading coefficient of the G polynomial; i.e., the scaling parameter for the controller output. He showed that if this parameter is much smaller than the true value, the other parameters will diverge. In contrast, if the parameter is much larger than the true value, the control action becomes very small and convergence of the parameters is very slow.

2.3.5 Extension to Multivariable case

The control structure derived for the SISO case can be extended to a class of multivariable systems by use of the feedforward terms. The system is decoupled by using feedforward action of the manipulated variable of one output as a measurable disturbance for the other outputs. This treatment will be restricted to systems with an equal number of inputs and outputs. In addition, the time delays due to

the interactive terms must be greater than the time delay due to the manipulated variable for any given loop.

Consider a multivariable system with M-inputs and M-outputs, the system can be described by the vector-matrix difference equation:

$$z^{-k_{ij}^A} [\underline{\underline{A}}] [\underline{y}_t] = z^{-k_{ij}^B} [\underline{\underline{B}}] [\underline{u}_t] + [\underline{\underline{C}}] [\underline{\varepsilon}_t] + z^{-k_{ij}^L} [\underline{\underline{L}}] [\underline{v}_t] \quad (2.48)$$

where $[\underline{\underline{A}}]$, $[\underline{\underline{B}}]$, $[\underline{\underline{C}}]$, $[\underline{\underline{L}}]$ are polynomial matrices in z^{-1} ; $[\underline{y}_t]$, $[\underline{u}_t]$, $[\underline{\varepsilon}_t]$ and $[\underline{v}_t]$ are vectors corresponding to the output, input, noise and feedforward disturbances; k_{ij}^A , k_{ij}^B , k_{ij}^L are the time delays of the output, input and feedforward disturbance for the j th component in the i th loop. The assumption is made that $[\underline{\underline{B}}_0]$ is nonsingular and that the roots of all the row polynomials in $[\underline{\underline{C}}]$ lie inside the z -domain unit circle.

The controller performance index is similar to the SISO case:

$$J = E\{ ([\underline{\underline{P}}][\underline{y}_{t+k}] - [\underline{\underline{R}}][\underline{w}_t])([\underline{\underline{P}}][\underline{y}_{t+k}] - [\underline{\underline{R}}][\underline{w}_t])^T \} + E\{ ([\underline{\underline{Q}}'][\underline{u}_t])([\underline{\underline{Q}}'][\underline{u}_t])^T \} \quad (2.49)$$

The difference is that $([\underline{\underline{P}}][\underline{y}_{t+k}] - [\underline{\underline{R}}][\underline{w}_t])$ is now a vector instead of a scalar value. As before, $[\underline{\underline{P}}]$, $[\underline{\underline{R}}]$, and $[\underline{\underline{Q}}']$ are matrix weighting functions in z^{-1} . Analogous with the SISO case, $([\underline{\underline{P}}][\underline{y}_{t+k}])$ can be replaced by the predicted vector $([\underline{\underline{P}}][\underline{y}_{t+k}])^*$ yielding the following equation:

$$J = E\{ (([\underline{\underline{P}}][\underline{y}_{t+k}])^* - [\underline{\underline{R}}][\underline{w}_t]) (([\underline{\underline{P}}][\underline{y}_{t+k}])^* - [\underline{\underline{R}}][\underline{w}_t])^T \} + E\{ ([\underline{\underline{Q}}'][\underline{u}_t])([\underline{\underline{Q}}'][\underline{u}_t])^T + [\underline{\varepsilon}_{t+k}][\underline{\varepsilon}_{t+k}]^T \} \quad (2.50)$$

where $[\underline{\varepsilon}_{t+k}]$ is a vector of weighted prediction errors which is assumed to be uncorrelated with all the inputs and outputs. Minimizing Equation (2.50) by setting $\partial J / \partial [\underline{u}_t]$ to zero gives:

$$\{([\underline{P}][\underline{y}_{t+k}])^* - [\underline{R}][\underline{w}_t]\}[\underline{B}_0]^T + [\underline{Q}][\underline{Q}_0][\underline{u}_t] = 0 \quad (2.51)$$

Defining:

$$[\underline{Q}] = [\underline{Q}][\underline{Q}_0]([\underline{B}_0]^T)^{-1} \quad (2.52)$$

the control law can now be written in terms of a predicted vector output as:

$$[\underline{\theta}^*_{t+k}] = [\underline{Q}][\underline{u}_t] + \{([\underline{P}][\underline{y}_{t+k}])^* - [\underline{R}][\underline{w}_t]\} \quad (2.53)$$

Replacing $([\underline{P}][\underline{y}_{t+k}])^*$ with an expression analogous to Equation (2.16) and using similar definition to those in Equations (2.29) to (2.33) and (2.37), the control law becomes:

$$z^{k_{ij}-k_{ij}}_{i \neq j} [\underline{G}][\underline{u}_t] = \frac{-1}{[\underline{Q}]} \{ z^{k_{ij}-k_{ij}}_{i \neq j} [\underline{F}][\underline{y}_t] + [\underline{H}][\underline{w}_t] + [\underline{D}][\underline{v}_t] \} \quad (2.54)$$

By expressing the interactive terms as feedforward disturbances, the parameters can be identified as a series of M loops. Each loop would correspond to the parameters for each row in Equation (2.54). Using recursive least squares identification, the algorithm for obtaining the parameters where i denotes the i th loop, is:

$$\theta_{i_t} = P_i \bar{y}_{i_t} - R_i \bar{w}_{i_t-k} B_{ii} + Q_i u_{i_t-k} B_{ii} \quad (2.55)$$

$$\hat{\underline{e}}_{i_t} = \hat{\underline{e}}_{i_{t-1}} + \underline{K}_{i_t} (\underline{e}_{i_t} - \underline{X}_{i_{t-k}}^T \underline{B}_{ii} \hat{\underline{e}}_{i_{t-1}}) \quad (2.56)$$

$$\underline{K}_{i_t} = \frac{\underline{P}_{i_t}^c \underline{X}_{i_{t-k}}^T \underline{B}_{ii}}{(\rho_i + \underline{X}_{i_{t-k}}^T \underline{B}_{ii} \underline{P}_{i_t}^c \underline{X}_{i_{t-k}})} \quad (2.57)$$

$$\underline{P}_{i_{t+1}}^c = 1/\rho_i * (1 - \underline{K}_{i_t}^T \underline{X}_{i_{t-k}} \underline{B}_{ii}) \underline{P}_{i_t}^c \quad (2.58)$$

By separating the loops, the identification scheme is simplified. Furthermore, the weighting functions for each loop is treated separately, that is the control effort weighting Q_i can be tuned individually loop by loop. In addition, this implementation allows the used of a different forgetting factor for each loop.

For most processes, the control law given by Equation (2.54) can be simplified since in practice one output in a multivariable system does not directly affect another output. It is the controller action that causes interactions. It follows that $[\underline{F}]$ is diagonal and the terms k_{ij}^A are zero. The simplified control law is then:

$$[\underline{G}][\underline{u}_t] = 1/[\underline{Q}] \{ [\underline{F}][\underline{y}_t] + [\underline{H}][\underline{w}_t] + [\underline{D}][\underline{v}_t] \} \quad (2.59)$$

A final note, although the identification is done one loop at a time, in general, the control signals have to be calculated simultaneously. The $[\underline{u}_t]$ vector is obtained directly by solving Equation (2.59). However, for systems with few inputs and outputs, the solution for $[\underline{u}_t]$ can be written out explicitly without too much complication so that $[\underline{u}_t]$ can be solved by direct substitution.

3. EXPERIMENTAL IMPLEMENTATION

3.1 DESCRIPTION OF EXPERIMENTAL EQUIPMENT

The pilot plant distillation column in the Department of Chemical Engineering at the University of Alberta is a 22.5 cm diameter column. It has a thermosyphon reboiler, a total condenser and eight bubble cap trays spaced 30.5 cm apart with four bubble caps per tray. A detailed description of the column and the associated equipment is given by Svrcek [2] and Pacey [6].

The feed is a water-methanol mixture containing 50% by weight methanol. The feed enters at a rate of 18.08 g/s and is separated to an overhead composition of 95% by weight methanol and a bottom composition of 5% by weight methanol. The control objective is to minimize variations in terminal compositions when the feed flow rate is changed by 25% from its normal steady state feed flow rate.

The distillation column is extensively instrumented to give maximum flexibility for studying different control strategies. A schematic diagram showing the extent of the instrumentation is given in Appendix A. All the measurable variables that affect the performance of the pilot plant column are recorded. Temperature measurements are taken for the liquids at every tray, the reboiler and condenser. As well, temperatures and flow rates of the bottoms, distillate, feed, reflux and steam are recorded. The cooling load for the condenser is monitored by measuring the

inlet and outlet temperatures and flow rate of the cooling water. The overhead composition is measured continuously using a capacitance probe while the bottom composition is obtained by using a Hewlett-Packard (HP) model 5720A gas chromatograph equipped with an automatic sampling valve. The sampling configuration is the one developed by Bilec [9]. The chromatogram is analyzed on-line by a HP 1000 mini-computer using a cycle time of three minutes. Additional variables that are measured include the column pressure and the liquid levels in both the condenser and reboiler. All the signals from the measured variables can be brought into the central HP-1000 mini-computer located in the Data Acquisition Control and Simulinion (DACS) center at the Department of Chemical Engineering for access by any program executing in the computer. It should be noted that the central computer is not the same one used for analyzing the gas chromatograph. Figure 3.1 shows how the distillation column is interfaced to the different computers in the DACS center.

The column is instrumented with local analog controllers that control all the flow rates, the condenser and reboiler levels, the feed and reflux temperatures and the column pressure. The instrumentation is such that control of the flows and temperatures can be taken over by the central HP 1000 mini-computer at the DACS center. The computer can assume direct digital control (DDC) or operate in the supervisory mode. With DDC, the computer sends the

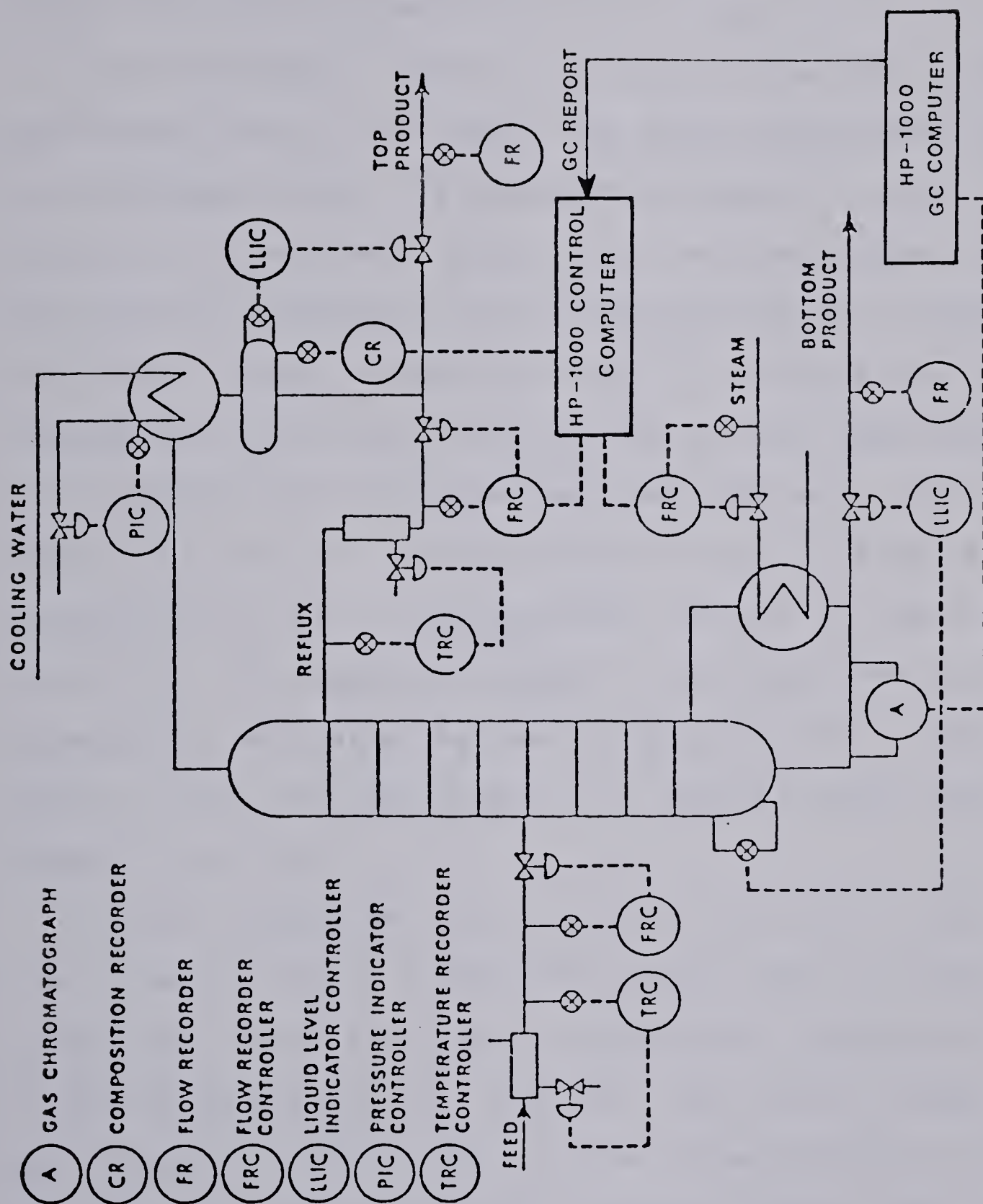


Figure 3.1 Schematic Diagram of the Distillation Column and Interface to Computers

control output to the valve directly. In the supervisory mode, the computer relays a set point to the local analog controller. The computer provides the 'master' loop and the analog controller becomes the 'slave' loop.

One advantage of supervisory control over DDC is that supervisory control can remove the undesirable effect caused by a nonlinear valve. If the control output is sent directly to a nonlinear valve, then the same change in valve position for different initial valve position will result in a different change in absolute flow. It follows that for the same deviation from different set points, the change in flow caused by the controller will be different. With supervisory control, the appropriate change in flow is calculated directly in the computer and sent to the flow controller. The required change in valve position will be subsequently determined by the analog controller. The result is that the same output will always produce the same change in the flow.

Another advantage with supervisory control is that the flow is held constant in between sample times so disturbances that affect the flow are immediately compensated for by the analog controller. With DDC, the control output is sent directly to the valve. If the valve position is held constant, pressure fluctuations between sample times can cause the flow to oscillate. Local control is advantageous in the event of computer failure.

However, supervisory control becomes less desirable

when the response time of the slave loop is of the same magnitude as the sample time of the master loop. In this situation, the dynamics of the slave loop adds directly to that of the master loop hence causing the control response to deteriorate.

For the pilot plant distillation column, the response time of the flow loops is much less than the sample time of the composition loops. Typically, the analog controllers will bring the flows to the new set points within a few seconds which is significantly less than the sample time for the composition loops. So consequently, the supervisory mode is used in the experimental study.

3.2 CONTROL LAW IMPLEMENTATION

To evaluate the self-tuning controller (STC), the output response using STC will be compared to the response achieved using a well tuned PID compensator. The comparison will be made on how well the controllers maintain the terminal compositions when the feed rate is changed.

For the three term proportional-integral-derivative compensator, the PID option in the Distributed System and Control (DISCO) package is used. Details of the implementation of DISCO on the pilot plant distillation column and the PID algorithm, written in the integral form, are given by Kan [66]. The STC algorithm could not be implemented using the DISCO executive because of size restriction. As a result, a completely separate program had

to be written to execute under the Real Time Executive operating in the HP 1000 mini-computer. A listing of the program is given in Appendix C. The STC algorithm used in the program is given by Equations (2.55) to (2.59).

4. SELECTION OF MANIPULATED VARIABLE

4.1 INTRODUCTION

Before undertaking the control study, the process output and control input pairs for the pilot plant distillation column were first chosen. The five variables that have to be controlled are: top composition, bottom composition, column pressure, level in the reflux drum and level in the reboiler. The five available manipulated variables are: reflux flow, distillate flow, bottom product flow, heat input to the reboiler and heat removal by the condenser. The variables feed temperature, reflux temperature and feed composition will be kept constant for this study.

There are various configurations for pairing the controlled and manipulated variables. The two types of strategies that have been used most frequently are the schemes termed material balance control and energy balance control [67]. With the energy balance scheme, the pressure of the column is controlled by manipulating the cooling load in the condenser. The two drum levels are maintained by changing the respective output flows. The top composition is controlled by the reflux flow while bottom composition is controlled by boilup rate. The term energy balance originates because the terminal compositions of the column are controlled by changing the energy balance of the system and it is clear that reflux flow and reboiler load changes

will directly affect the energy balance of the column.

With material balance control, the terminal composition is controlled by directly altering the material balance of the distillation column. One of the product flow rate is manipulated to control its own composition. Level for that end of the column is then controlled by either reflux flow or reboiler load depending on which product flow rate is used. The remaining variables will be paired the same way they are matched in the heat balance strategy.

Distillate flow is usually selected to control the material balance. The reason for this is that problems could develop if the the reboiler level is to be controlled by manipulation of the reboiler load. As the distillate flow is usually used, the ensuing discussion on material balance control will be based on using distillate flow to control the top composition. Analogous arguments can be given that will be applicable to the case when the bottom composition is controlled by manipulating bottom product flow rate. Schematic diagrams of the overhead composition control configuration for the material and energy balance control schemes are shown in Figures 4.1 and 4.2.

The concept of 'floating' the reflux flow on level control is not an easy one to accept since basic to the distillation column design procedure is the assumption of the operation at a constant reflux flow and fixed reflux ratio. Nevertheless, studies on different distillation columns have shown that the use of the material balance

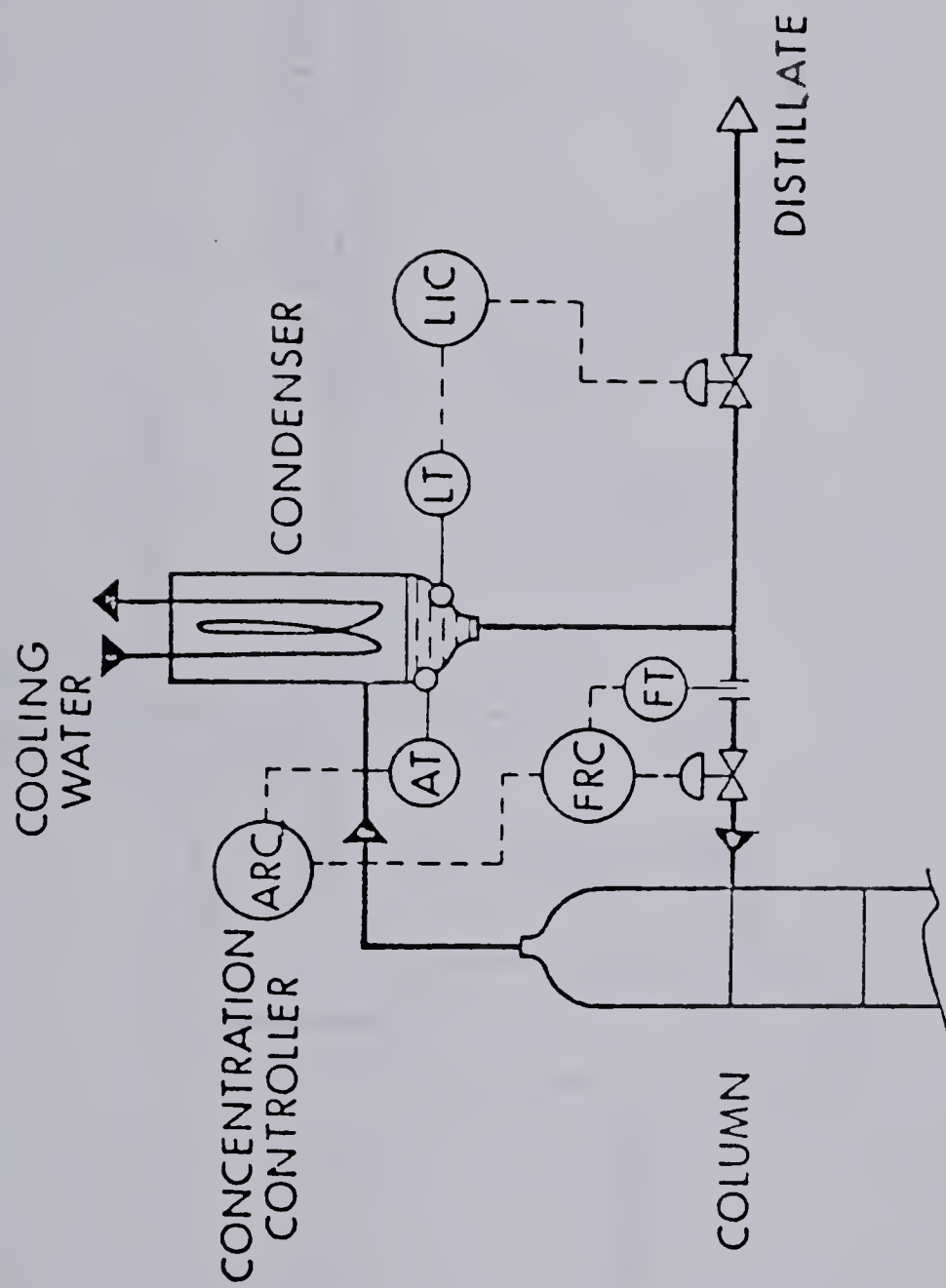


Figure 4.1 Schematic Diagram of the Energy Balance Control Scheme

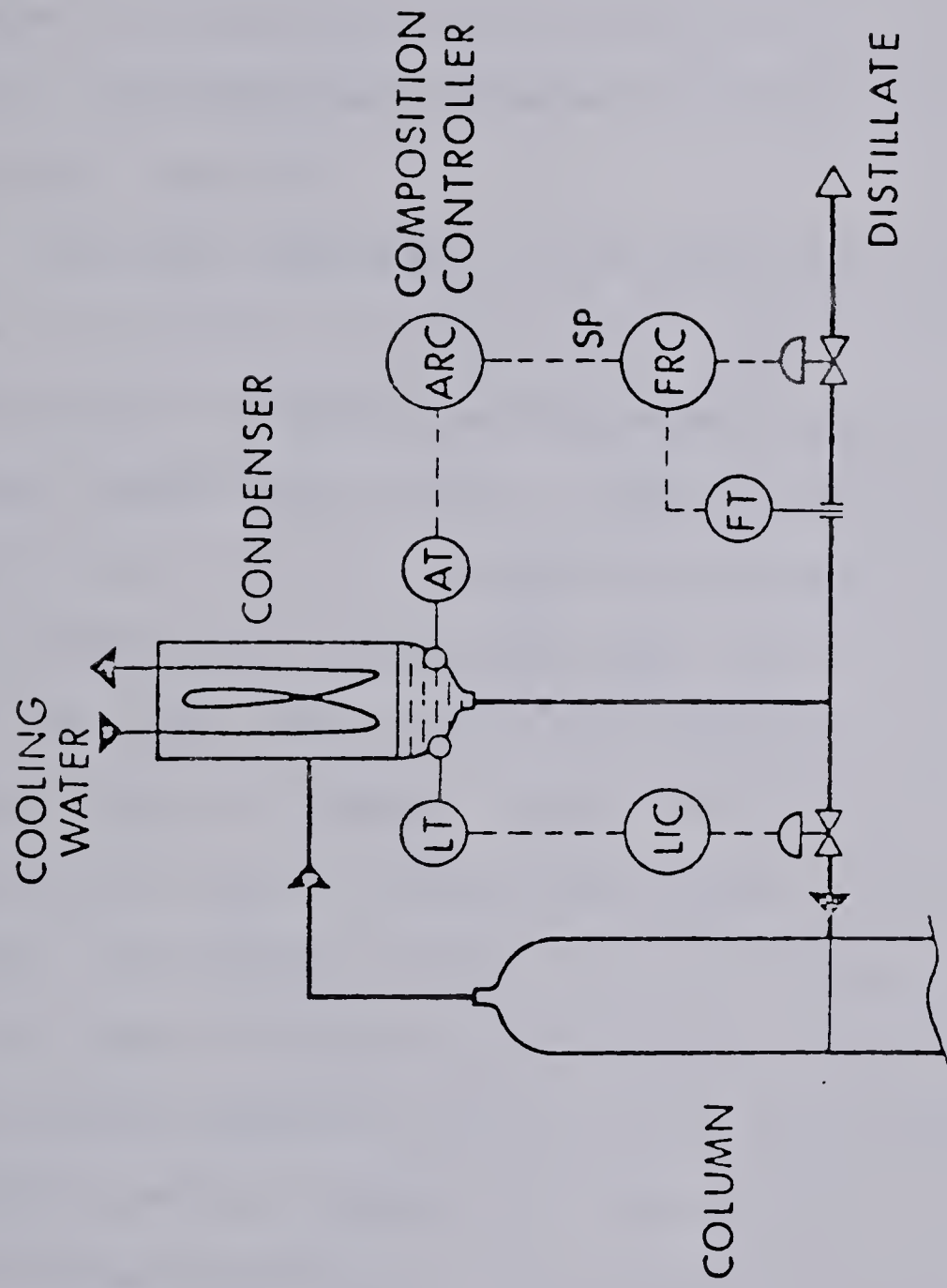


Figure 4.2 Schematic Diagram of the Material Balance Control Scheme

control scheme can result in improved control behaviour over the case when energy balance control is used [68-73]. In certain cases, it has been found that it is not possible to stabilize control without using the material balance approach [72,73]. In both cases, control of the overhead composition could not be realized by direct manipulation of reflux flow because of the very high external reflux ratio (greater than 25).

The main advantage of the material balance strategy is that it exhibits inherent reflux control when an energy disturbance is encountered because the reflux flow readjusts itself before any action is taken by the composition controller. This is accomplished because an energy disturbance will change the vapor rate in the column. As the new vapor rate reaches the condenser, the reflux drum level starts to change. Since the reflux flow is used to control the level, it will be adjusted to maintain level set point. The overall result is that reflux flow will change in the same direction as the change in vapor rate. Since the scheme inherently handles heat disturbances, material balance control reduces the interaction that occurs during simultaneous control of top and bottom composition. Variations in the reboiler load due to the bottom composition controller are treated as heat disturbances by the overhead controller so the reflux flow will be adjusted in the appropriate direction before any change in overhead composition is detected.

Other advantages of the material balance scheme are the following:

- a. For the case where the reflux ratio is very large, using the distillate flow to control the reflux drum level can be very difficult. Since the distillate flow rate is much lower than the reflux flow, a large percentage change in distillate flow is required to compensate for small percentage changes in reflux flow. These large swings in distillate flow can be a big problem especially in situations where the distillate product is a feed to another column. Additional drum capacity might be needed to damp out the variation.
- b. The material balance strategy is readily adapted for feedforward application since the required distillate to feed ratio can be determined exactly from the material balance relationship. Thus, the necessary change in distillate flow to compensate for changes in feed flow rate and feed composition can be easily calculated and subsequently fed forward through the appropriate dynamic compensator.
- c. The material balance scheme will not exhibit a transient inverse response that is sometimes observed when reflux flow is directly used to control overhead composition. The inverse response as explained by Rosenbrock [74] is caused by plate

overflow when an increased vapor rate first reaches the top of the distillation column. The overflow acts as an increased reflux flow hence causing the composition to initially become richer in the more volatile component. When the overflow stops, the increased vapor rate takes over and the composition becomes leaner in the more volatile component. With material balance control, the increased vapor rate would cause the reflux to increase and as a result, the inverse response should not occur.

The main disadvantage of the material balance strategy over the heat balance control case is the addition of the control dynamics of the level loop into the dynamics of the composition loop. When the top composition varies from the set point, the distillate flow is first adjusted. This change in turn affects the level in the reflux drum. Subsequently, the reflux flow then adjusts to maintain the drum level. The composition will not begin to change until the reflux flow has been changed. The overall result is that more dynamics is added hence slowing the response of the overhead composition loop. Furthermore, controller response to set point changes could be slower than the case when the reflux is adjusted directly.

4.2 ANALYSIS OF INTERNAL REFLUX RATIO SENSITIVITY TO HEAT DISTURBANCE

The self-regulation of the reflux flow in material balance control will lead to improved control performance only if the change in reflux flow is changed in a way which maintains the internal reflux ratio. Clearly, it is the maintenance of the internal reflux ratio that allow the material balance scheme to be less sensitive to heat disturbances. By holding a constant internal reflux ratio, the slope of the operating curve remains unchanged hence the composition is unaltered.

For a distillation column operating with a large external reflux ratio, a material balance strategy will maintain the internal reflux ratio better than the energy balance case. However, if the external reflux ratio is low, the self-regulating change in reflux flow can cause the internal reflux ratio to vary more than the energy balance scheme. The best way to illustrate the effect is a simple numerical example. It should be noted that in the ensuing analysis, the V/L ratio will be considered instead of L/V traditionally defined as the internal reflux ratio. The reason for this is that the region of interest is for internal reflux ratio less than 1. The reciprocal, V/L , will have a greater numerical range.

Consider a situation where the external reflux ratio is 10.0, that is:

$$\frac{\text{Reflux Flow (L)}}{\text{Distillate Flow (D)}} = 10.0 \quad (4.1)$$

or,

$$L = 10.0D \quad (4.2)$$

Since the vapor rate (V) is given by:

$$V = L + D \quad (4.3)$$

Combining Equation (4.2) and (4.3) yields:

$$V = 11.0D \quad (4.4)$$

and the resulting ratio is:

$$\frac{V}{L} = \frac{11.0D}{10.0D} = 1.1 \quad (4.5)$$

Suppose that the column is subjected to energy disturbances that correspond to a required change in the vapor rate of a 20% increase and a 20% decrease. The required new V/L ratio for both material balance and energy balance strategies for such changes are given in Table 4.1. Recall that for material balance control, the reflux flow will change by the same amount the vapor rate is changed. From Equation (4.4), a 20% change in vapor rate is:

$$0.2V = 0.2 * 11.0D = 2.2D \quad (4.6)$$

It is evident from the values in Table 4.1 that the material balance control clearly maintains a more constant V/L ratio than the heat balance case as the ratio under the material scheme changes less than 3% compared to the 20%

changes for the energy balance case.

Table 4.1 Variation of V/L with Control Scheme for $\pm 20\%$ Changes in Vapor Rate at an External Reflux Ratio of 10.0

Control Scheme	Positive Disturbance	Deviation From S.S.	Negative Disturbance	Deviation From S.S.
Material Balance	$\frac{11.0+2.2}{10.0+2.2} = 1.08$	-1.6%	$\frac{11.0-2.2}{10.0-2.2} = 1.13$	+2.6%
Energy Balance	$\frac{11.0+2.2}{10.0} = 1.32$	+20.0%	$\frac{11.0-2.2}{10.0} = 0.88$	-20.0%

Now, consider the case when the external reflux ratio is only 1.0, that is:

$$\frac{L}{D} = 1.0 \quad (4.7)$$

so,

$$V = L + D = 2.0D \quad (4.8)$$

and the steady value of V/L is:

$$\frac{V}{L} = \frac{2.0D}{D} = 2.0 \quad (4.9)$$

The new V/L ratios that result from $\pm 20\%$ changes in vapor rate are given in Table 4.2.

Table 4.2 Variation of V/L with Control Scheme for $\pm 20\%$ Changes in Vapor Rate at an External Reflux Ratio of 1.0

Control Scheme	Positive Disturbance	Deviation From S.S.	Negative Disturbance	Deviation From S.S.
Material Balance	$\frac{2.0+0.4}{1.0+0.4} = 1.71$	-14.3%	$\frac{2.0-0.4}{1.0-0.4} = 2.67$	+33.3%
Energy Balance	$\frac{2.0+0.4}{1.0} = 2.40$	+20.0%	$\frac{2.0-0.4}{1.0} = 1.60$	-20.0%

As can be seen from the numerical values in Table 4.2, which are markedly different from those in Table 4.1, the material balance control scheme is preferable for a positive disturbance but just the reverse for the negative disturbance. Clearly, for a situation where the external reflux ratio is 1.0, the energy balance strategy will prove to be less sensitive to heat disturbances. The choice of energy balance control becomes even more advantageous if it is recalled that with the energy balance scheme, the dynamics of the reflux drum level do not enter the overhead composition control loop.

These simple numerical calculations have clearly

illustrated that the material balance control scheme will not always be superior to heat balance control with the choice of scheme dependent on the external reflux ratio. While it is true that most industrial columns operate with an external ratio much greater than one, that is not necessarily the case with pilot plant distillation columns. In this study, the pilot plant column operating conditions are:

$$\begin{aligned}\text{Reflux Flow} &= 10.2 \text{ g/s} \\ \text{Distillate Flow} &= 8.8 \text{ g/s}\end{aligned}$$

so, the steady state V/L ratio is:

$$\frac{V}{L} = \frac{10.2 + 8.8}{10.2} = 1.87$$

The change in the V/L ratio for $\pm 20\%$ changes in the vapor rate are shown in Table 4.3.

Table 4.3 Variation of V/L with Control Scheme for $\pm 20\%$ Changes in Vapor Rate

Control Scheme	Positive Disturbance	Deviation From S.S.	Negative Disturbance	Deviation From S.S.
Material Balance	$\frac{19.0+3.8}{10.2+3.8}=1.63$	-12.6%	$\frac{19.0+3.8}{10.2-3.8}=2.34$	+27.6%
Energy Balance	$\frac{19.0+3.8}{10.2}=2.24$	+20.0%	$\frac{19.0-3.8}{10.2}=1.49$	-20.0%

The results presented in Table 4.3 indicate that the reduced deviation with the material balance control compared to energy balance control for a positive disturbance (7.4% lower deviation) is just about equal to the higher deviation of 7.6% for the negative disturbance. As both strategies are equally sensitive to heat disturbances, heat balance control is the strategy chosen because the material balance scheme has the added disadvantage of having a slower response due to the inclusion of the condenser level dynamics.

4.3 EXPERIMENTAL OPEN LOOP RESPONSES

To substantiate the analysis carried out in the previous section, experimental open loop responses were obtained for the top and bottom compositions to changes in steam flow using both the material balance and energy balance configurations. The steam flow changes simulated heat disturbances entering the system.

For each control strategy, steam flow rate was increased and decreased to flow rates representing a change of 7 1/2% from its steady state value. The open loop responses for the material balance scheme are shown in Figures 4.3 and 4.4. The corresponding responses for energy balance control are shown in Figures 4.5 and 4.6. The results are summarized in Table 4.4.

Table 4.4 Top and Bottom Composition Changes to Steam Flow Rate Disturbances.

Control Scheme	7 1/2% Increase In Steam Flow Rate		7 1/2% Decrease in Steam Flow Rate	
	XD Change	XB Change	XD Change	XB Change
Material Balance	+0.82%	-0.55%	-1.32%	+1.32%
Energy Balance	-1.50%	-4.08%	+0.58%	+6.57%

The results of Table 4.4 are in agreement with the predictions given by the analysis of the vapor-to-liquid ratio for vapor rate changes. The values in Table 4.3 indicate that the overhead composition should be more sensitive to a positive energy disturbance when the energy balance scheme is used than if material balance control is used. This is indeed the case as can be seen from the 1.5% decrease in overhead composition with energy balance control compared to the 0.82% increase under the material balance scheme. The results for the negative disturbance are also consistent with those in Table 4.3 since the top composition decreases by 1.32% for material balance control but only increases by 0.58% for energy balance control.

The potential advantage of the material balance scheme for systems with a large external reflux ratio is illustrated by considering the bottom composition. Since the entering feed is a liquid, the liquid rate below the feed

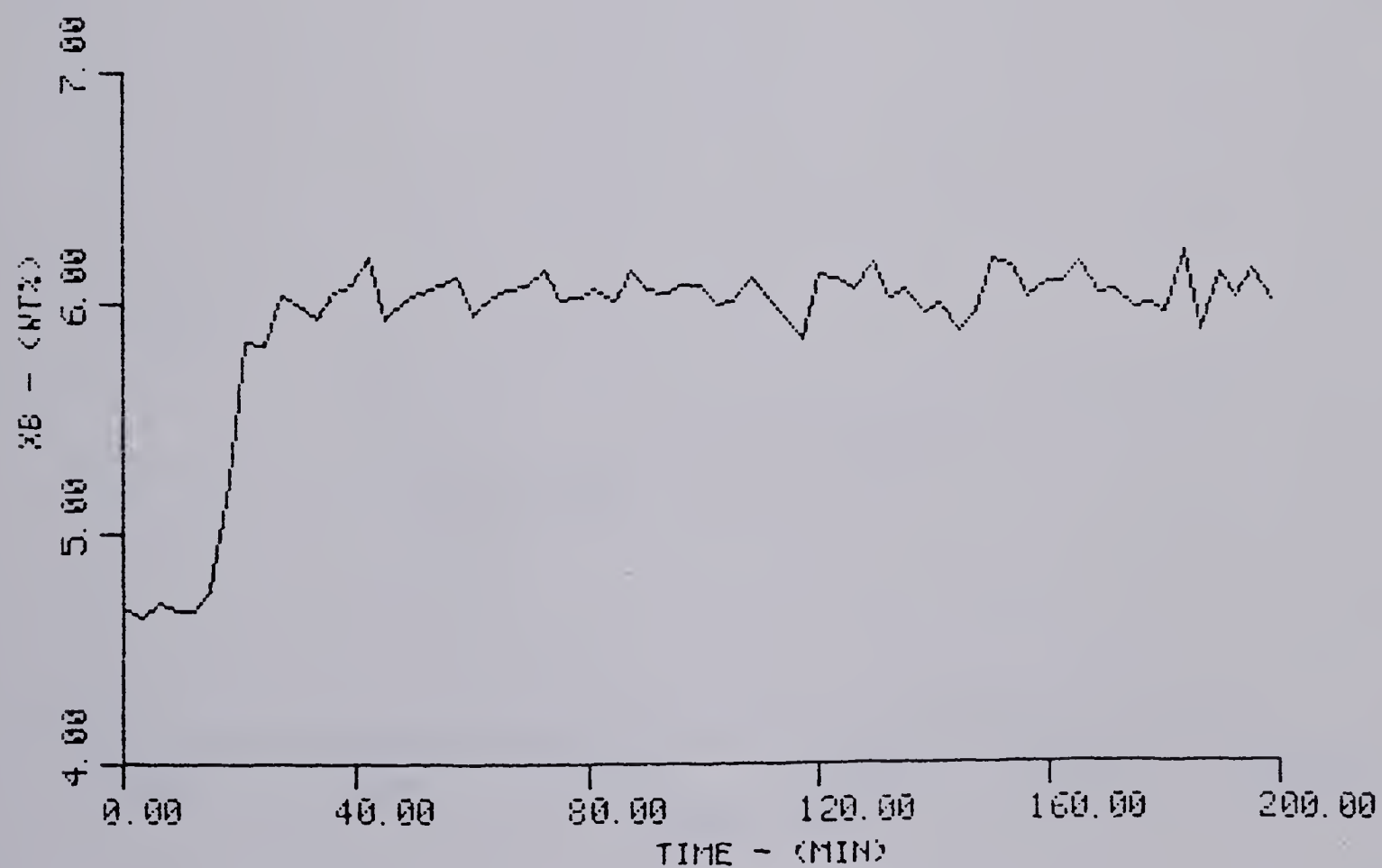
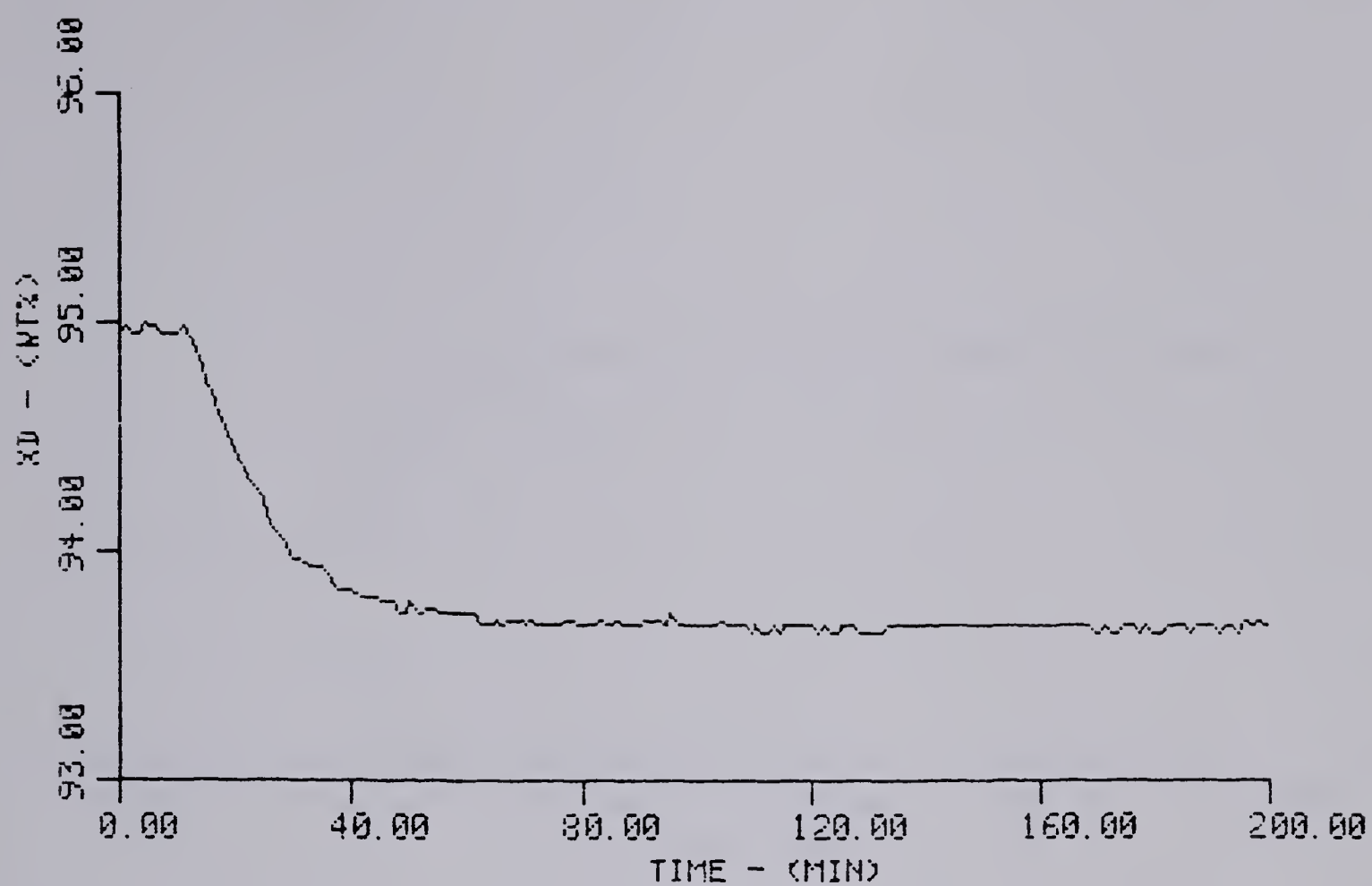


Figure 4.3 Open Loop Response to a 7 1/2% Step Decrease in Steam Flow With A Material Balance Configuration

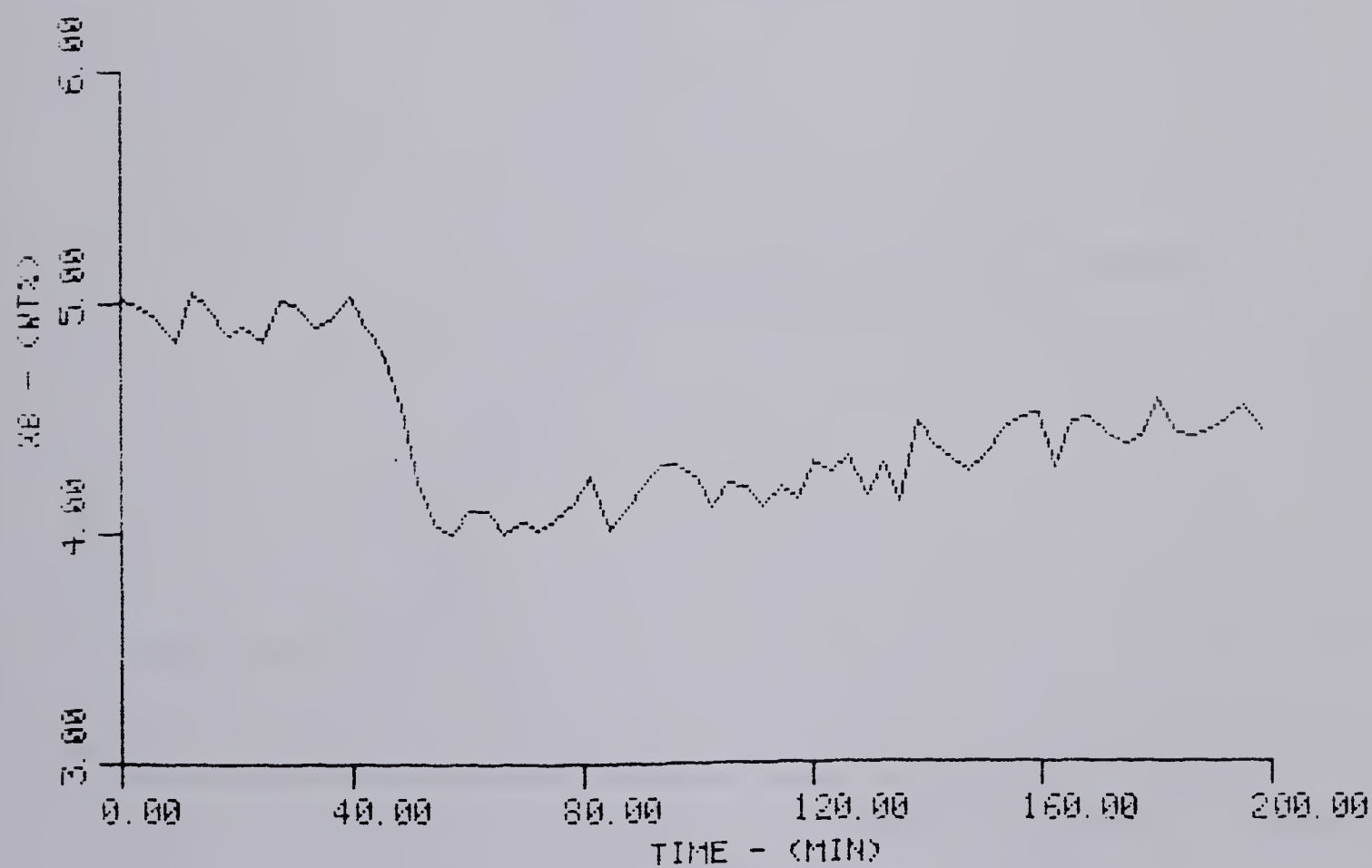
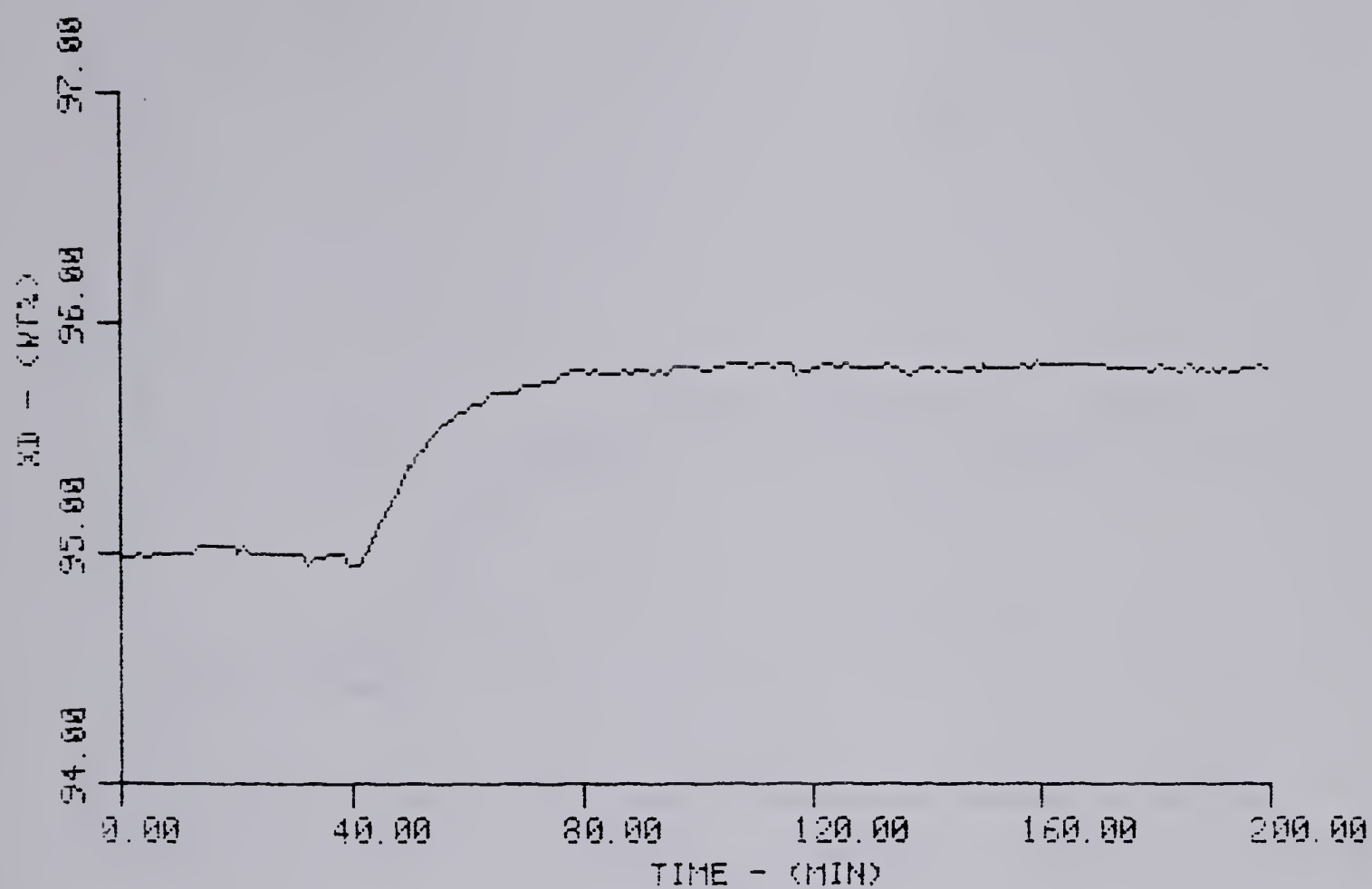


Figure 4.4 Open Loop Response to a 7 1/2% Step Increase in Steam Flow With A Material Balance Configuration

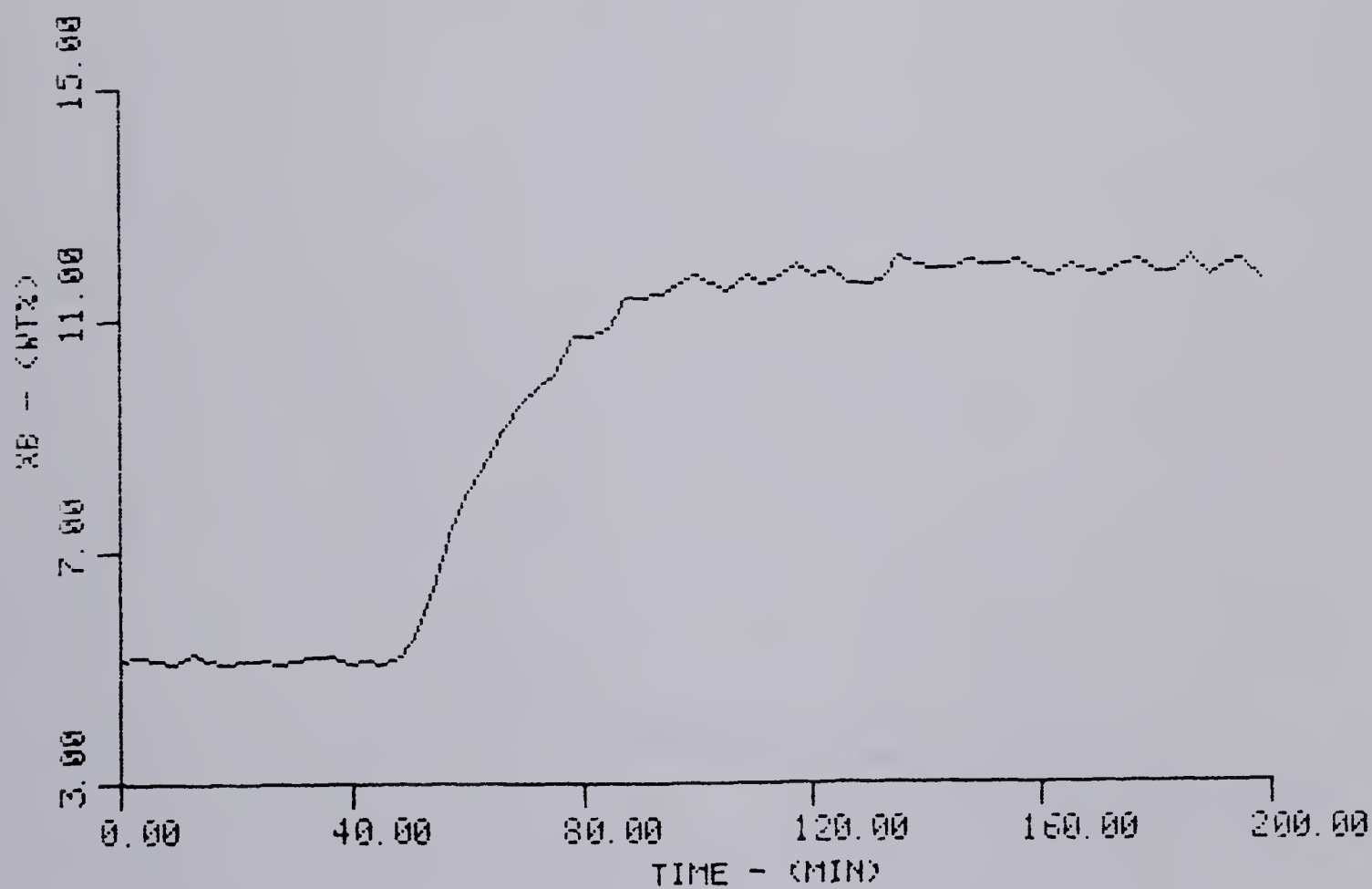
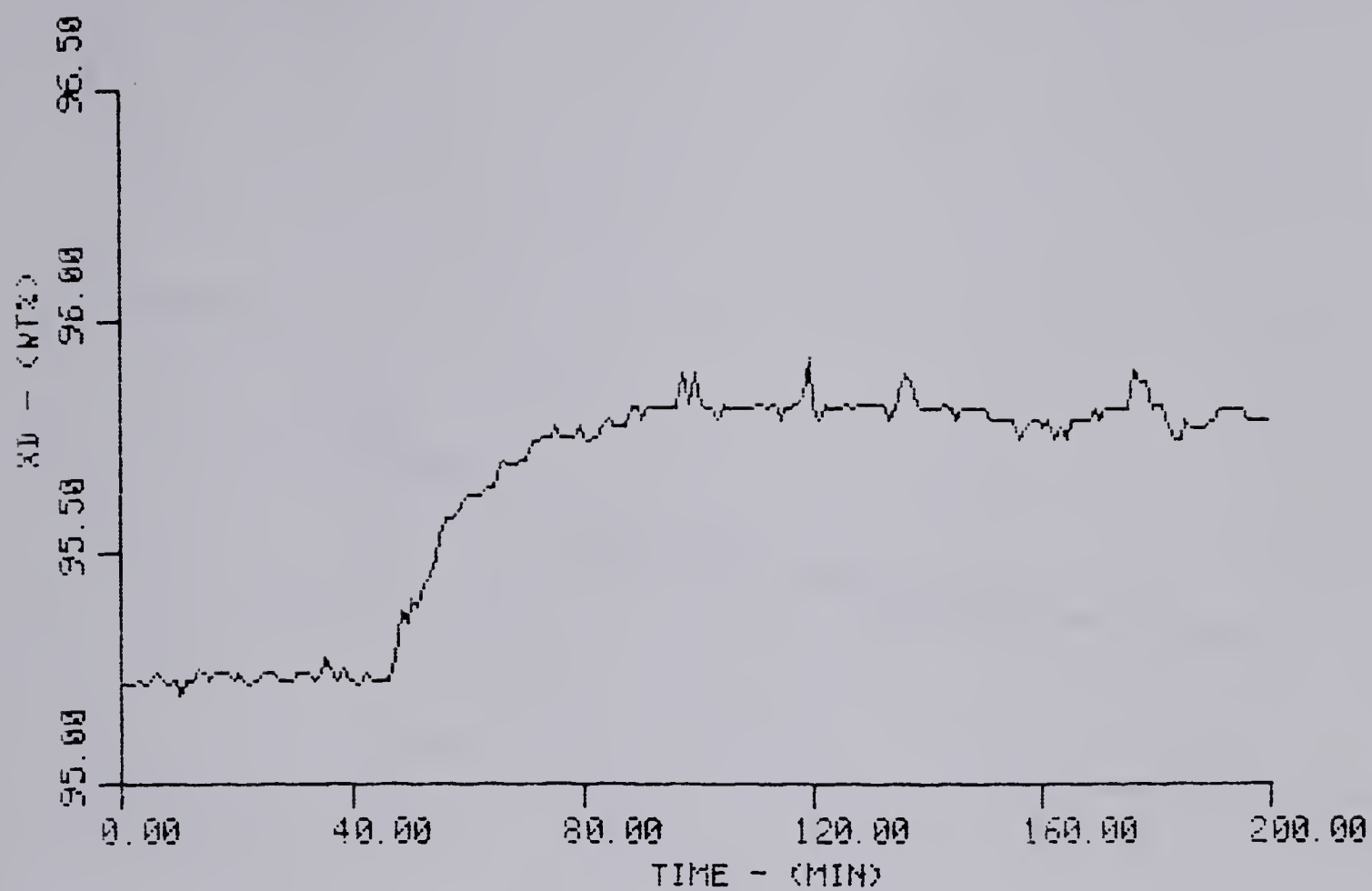


Figure 4.5 Open Loop Response to a 7 1/2% Step Decrease in Steam Flow With an Energy Balance Configuration

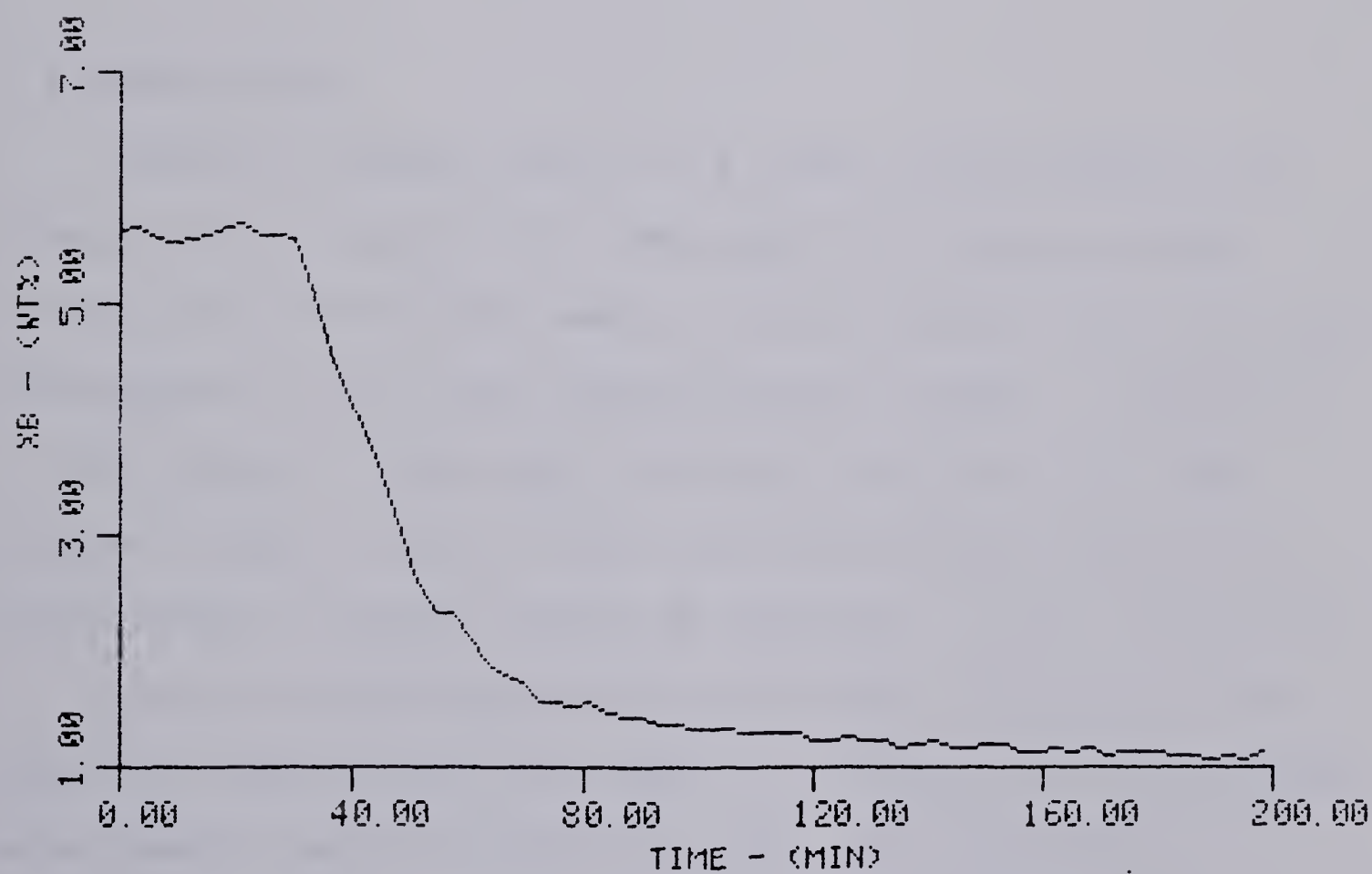
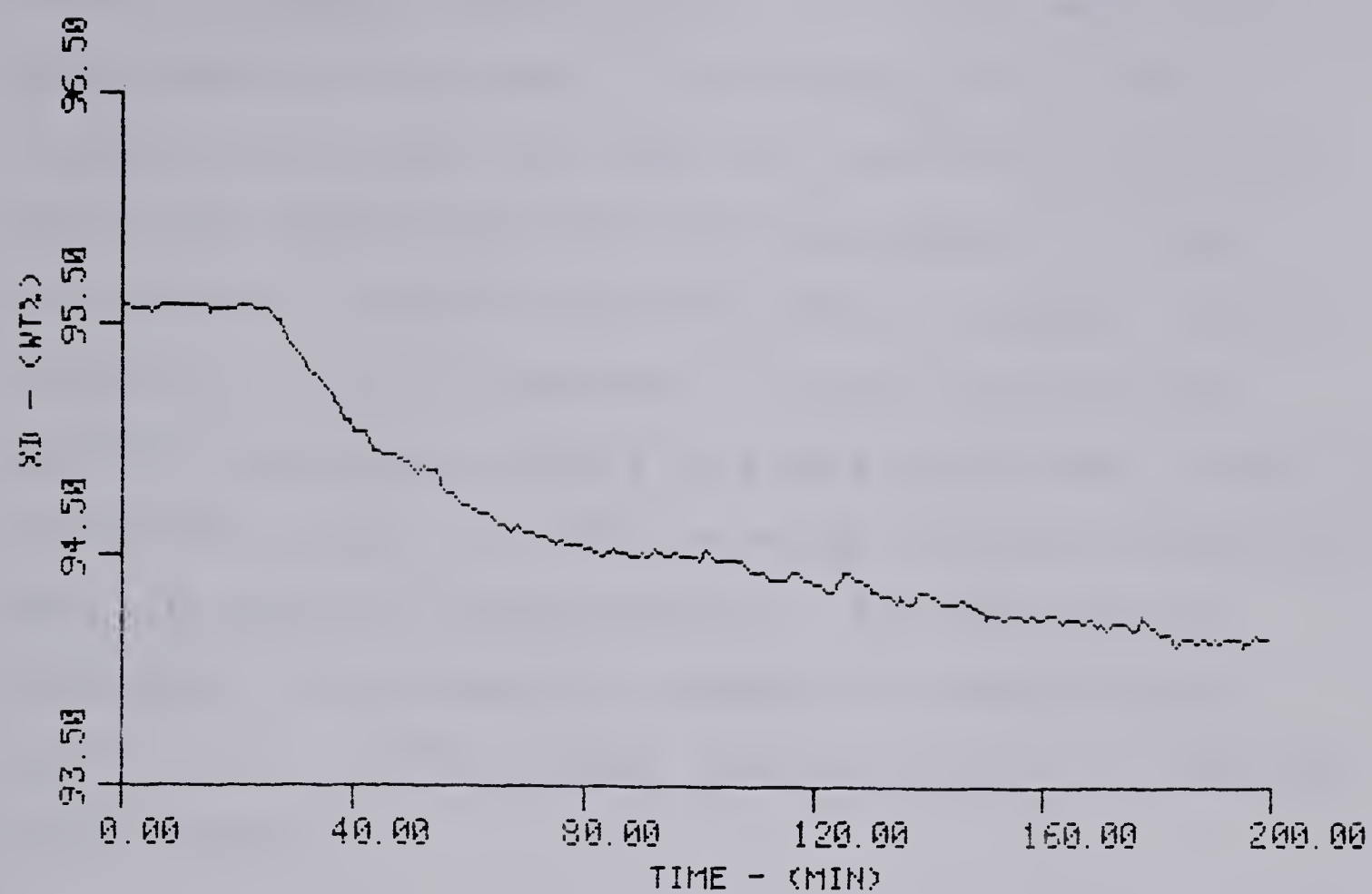


Figure 4.6 Open Loop Response to a 7 1/2% Step Increase in Steam Flow With an Energy Balance Configuration

plate is increased substantially. In effect, more reflux is being added to the column. From the results of Table 4.1, it should follow that the vapor-to-liquid ratio and hence the bottom composition should be less sensitive to heat disturbances if material balance control is used. This is in fact the situation observed. For both positive and negative disturbances, there is a more significant change in the bottom composition with the energy balance configuration than with material balance control. For the positive disturbance, the composition change with energy balance control is 7 1/2 times larger than the change with material balance control.

4.4 CONCLUSION

Material balance control is shown to be superior to energy balance control for the case of a large external reflux ratio while with smaller reflux ratio it will be more advantageous to use the energy balance scheme. Comparison of the change in the vapor-to-liquid ratio for the same change in vapor rate for both control schemes can help to decide which strategy should be used for a given situation.

For the column operating conditions, analysis of the vapor-to-liquid ratio for vapor rate changes and subsequent experimental verification have shown that the choice of control strategy depends on the type of disturbance. Energy balance control is less sensitive to a negative energy disturbance but because the material balance control scheme

inherently includes condenser level dynamics, the energy balance scheme is employed in this study.

5. EXPERIMENTAL RESULTS

5.1 INTRODUCTION

This chapter presents the results of the experimental runs that have been carried out on the distillation column using the PID and self-tuning controllers. A brief discussion of the experimental procedures and the actual programs that were used will be given. The results are presented in the following order:

- a. PID runs for top composition, bottom composition and dual composition control;
- b. STC runs for top composition control;
- c. STC runs for bottom composition control;
- d. STC runs for dual composition using multiloop configuration (ML-STC);
- e. STC runs for dual composition using multivariable configuration (MV-STC).

Control of top product composition will be used to study most of the properties of the self-tuning controller while bottom composition will be used to evaluate the feedforward (FF) feature of the STC.

5.2 EXPERIMENTAL PROCEDURE

Four types of disturbances were used to perturb the distillation column:

- a. a 25% reduction in feed flow rate;
- b. a subsequent feed flow rate increase to return the

- rate to its normal steady state flow;
- c. a 25% increase in feed flow rate;
- d. a subsequent feed flow rate decrease to return the rate to its normal steady state value.

For the study of top composition control, the disturbances of a 25% increase in feed flow rate and the subsequent return of the flow rate to its steady state value were not used because the top composition showed virtually no change for these disturbances as can be seen from the responses in Figures 5.1 and 5.2 (The arrow in the figures indicate the time that the step change in feed flow rate is introduced). In addition, only a limited number of multiloop and multivariable runs were performed for these two disturbances because a condition of flooding nearly resulted when the feed rate was increased 25% above the steady state flow rate.

With single loop control, the self-tuning controller was started with zero initial parameter estimates and the initial $\underline{\underline{P}}^C$ matrix is set at $1000\underline{\underline{I}}$ to show that there was lack of confidence in the initial estimates. The controller was given time to obtain reasonable estimates before a feed disturbance was introduced. For the multiloop and multivariable configuration, the self-tuning controller was started with the final estimates obtained from the single loop control tests. To show that there was confidence in these estimates, the initial $\underline{\underline{P}}^C$ matrix was set equal to $.01\underline{\underline{I}}$. As with the single loop control case, the parameters

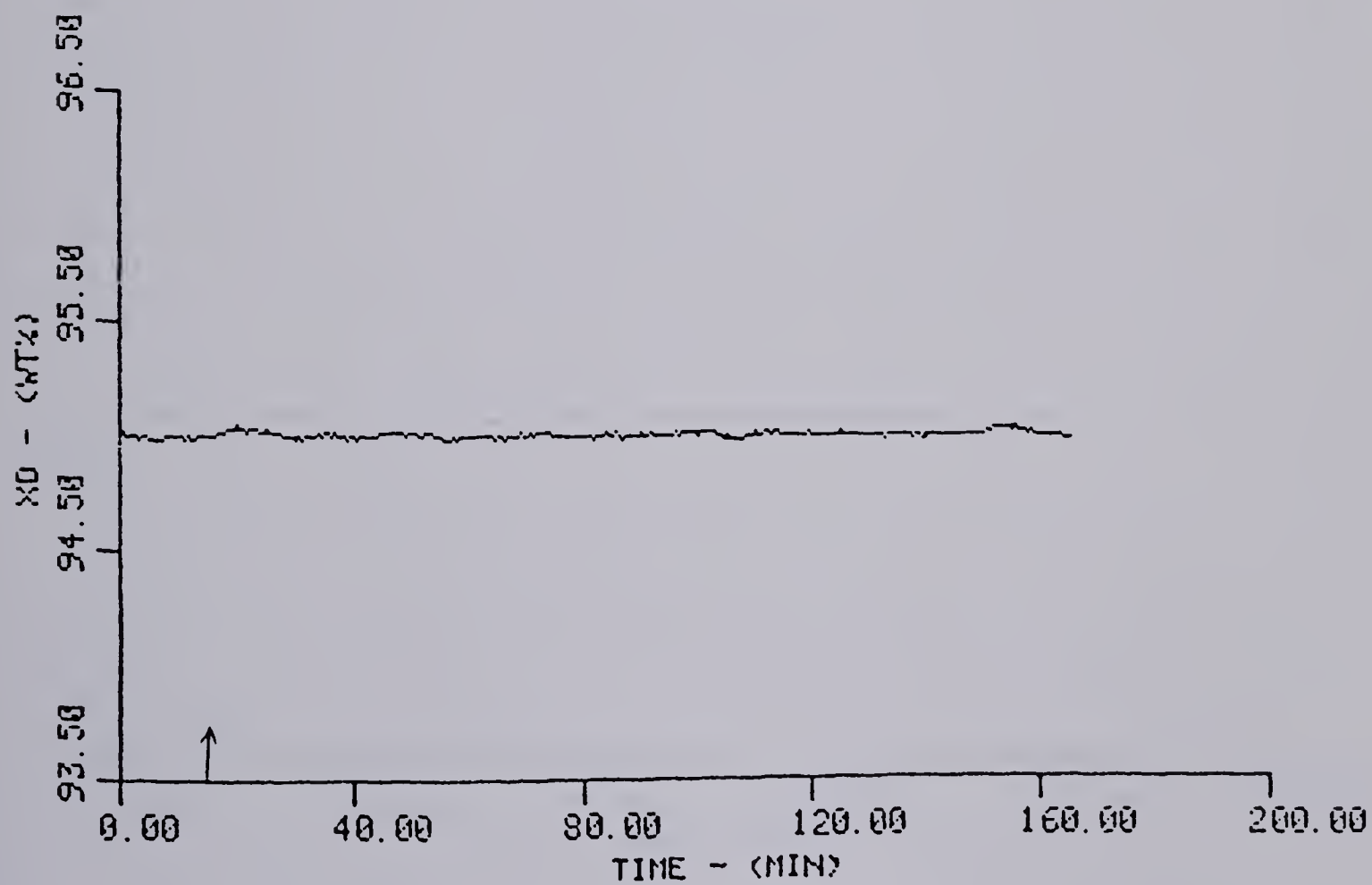
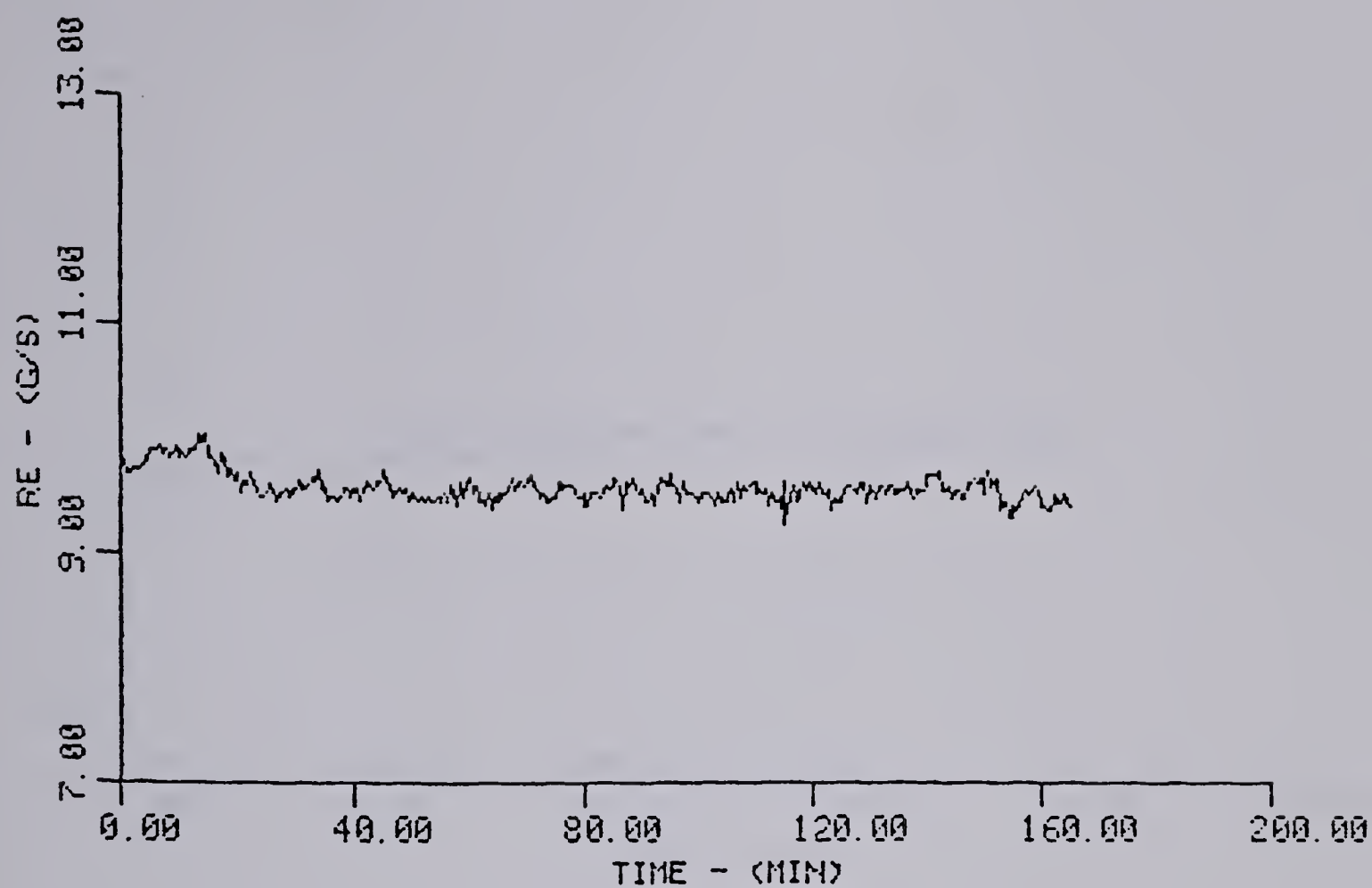


Figure 5.1 PID Control of Top Composition for a +25% Step Change in Feed Flow Rate

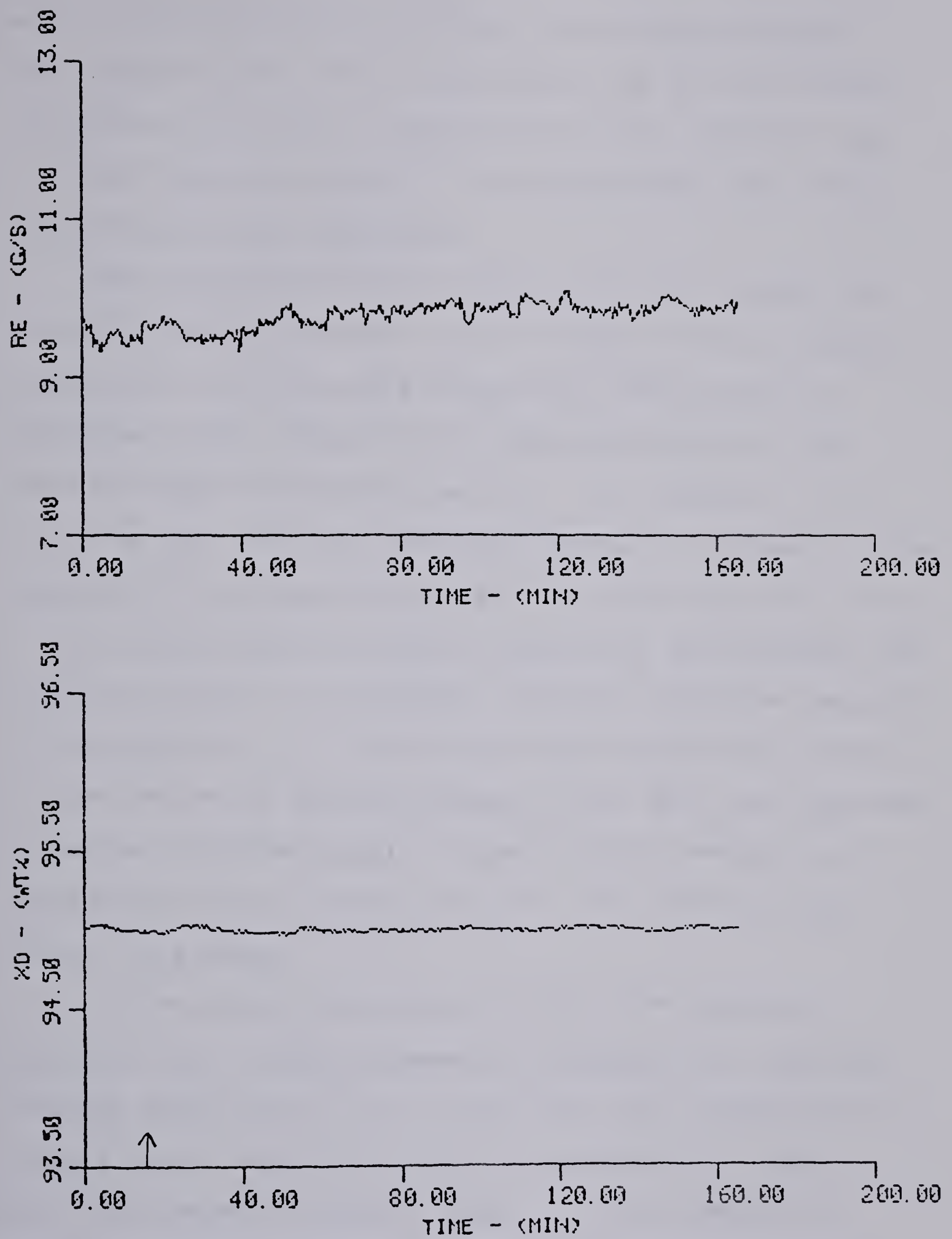


Figure 5.2 PID Control of Top Composition for a Step Decrease in Feed Flow Rate to its Normal Steady State Value

were allowed to converge before a feed change was made. When comparing the control performance of the self-tuning controller with that of the PID controller, only STC runs for which the disturbance is introduced after the initial identification are considered

The discussion given in Section 2.3.1 has shown that the self-tuning controller can be thought of as a feedback controller with a discrete compensator which uses the predicted output instead of the measured output as the feedback signal for control action. From Equation (2.23), it can be seen that the discrete compensator is equal to the inverse of the Q-weighting (Q-WT). By selecting the inverse of the Q-weighting to be the PI controller and assuming the predicted output to be correct, standard techniques proposed by Verbruggen et al. [75] can be used to obtain an initial choice for the PI constants based on the open loop response. Since the predicted output is used, the PI constants are those based on the system except for the effect of the process time delay.

For the single loop control tests, the constants obtained from the method based on the open loop responses produced good control but it was found that increasing the integral gain improved the control response. For the multiloop and multivariable cases, a certain amount of de-tuning was necessary to prevent oscillatory responses.

Experience with tuning the Q-weighting showed that the response was not as sensitive to variation in the

Q-weighting parameters compared to the sensitivity of the PID compensator to changes in the PID constants. The reason for this insensitivity is due to the fact that the parameters change to compensate to some degree for changes in the Q-weighting constants. Although the sensitivity was different, it was found that the PI constants in the Q-weighting could be tuned in the same way a normal PI compensator is tuned; i.e., the relationship between the proportional and integral gains and how they affect the output response remains the same.

5.3 COMPUTER PROGRAMS

With the change of computer system in the DACS center, support programs for the distillation column had to be written for the new system. A monitor program, BDC99, was written to monitor key process variables on the column. When the values of the variables exceeds specified constraints, BDC99 will shut down the distillation column.

As discussed in chapter three, the chromatogram analysis is done by a computer separate from the central computer where the other process measurements are available. As the control algorithm will execute in the central computer, a program, GCLNK had to be written to receive the report from the GC computer and transfer the information into some buffer in the central computer. Programs GCSCCH and GCSWT then processed the information in the buffer to extract the value of the the bottom composition from the

report. The programs also served to start the GC sample cycle at the appropriate time.

The implementation of the self-tuning control algorithm on the column was carried out with programs STC and HSET. Program HSET is used to alter the data file that contains the starting parameters and options used for the self-tuning controller while program STC contains the control algorithm. In addition to the algorithm itself, program STC takes care of storing the relevant variables in a disk file. When the STC program is executing, communication with the program can be accomplished through a group of digital switches. Utility programs HPLLOT and ERSUM were written to provide plots and calculation of the sum of the absolute error (SAE) of the results stored in the data files for a user specified period of time (number of sampling intervals).

A listing of programs HSET and STC is included in Appendix C. A listing of the remaining programs is only included in certain copies of this work.

5.4 CONTROL BEHAVIOUR USING PID CONTROL

The control performance obtained using conventional PID control action will be used as the base case for evaluation of the self-tuning controller. For the three cases of: top composition control, bottom composition control and simultaneous product composition control, the PID results presented are for a well tuned PID compensator. While the performance cannot be guaranteed to have been obtained using

true optimal tuned PID controller settings, the constants have been so well tuned that any changes in their values cause the control performance to deteriorate. The controller performance is measured by the sum of the absolute error (SAE). For top composition control, with a sampling time of 1 minute, the SAE is taken over one hour or 60 samples after the feed flow rate change. For bottom and dual composition control, with a sampling time of 3 minute, the SAE is taken over 1 1/2 hour period or 30 samples after the feed flow rate change.

For the single loop control test, the first choice of PID constants was obtained using the techniques proposed by Verbruggen et. al. [75] which bases the selection of PID constants on the system open loop response. Using these values as a starting point, the parameters were altered in a systematic way until further changes in the constants resulted in deterioration of the control performance. For the simultaneous control case, the well tuned single loop constants, reduced by one half were used as the initial choice. The constants were reduced because of the interaction between the top and bottom composition control loops.

The experimental results for top composition control are those given in Section 5.2 (cf. Figures 5.1 and 5.2) and in Figures 5.3 and 5.4. The bottom composition control results are shown in Figures 5.5 to 5.8. For bottom composition control, the PID constants are significantly

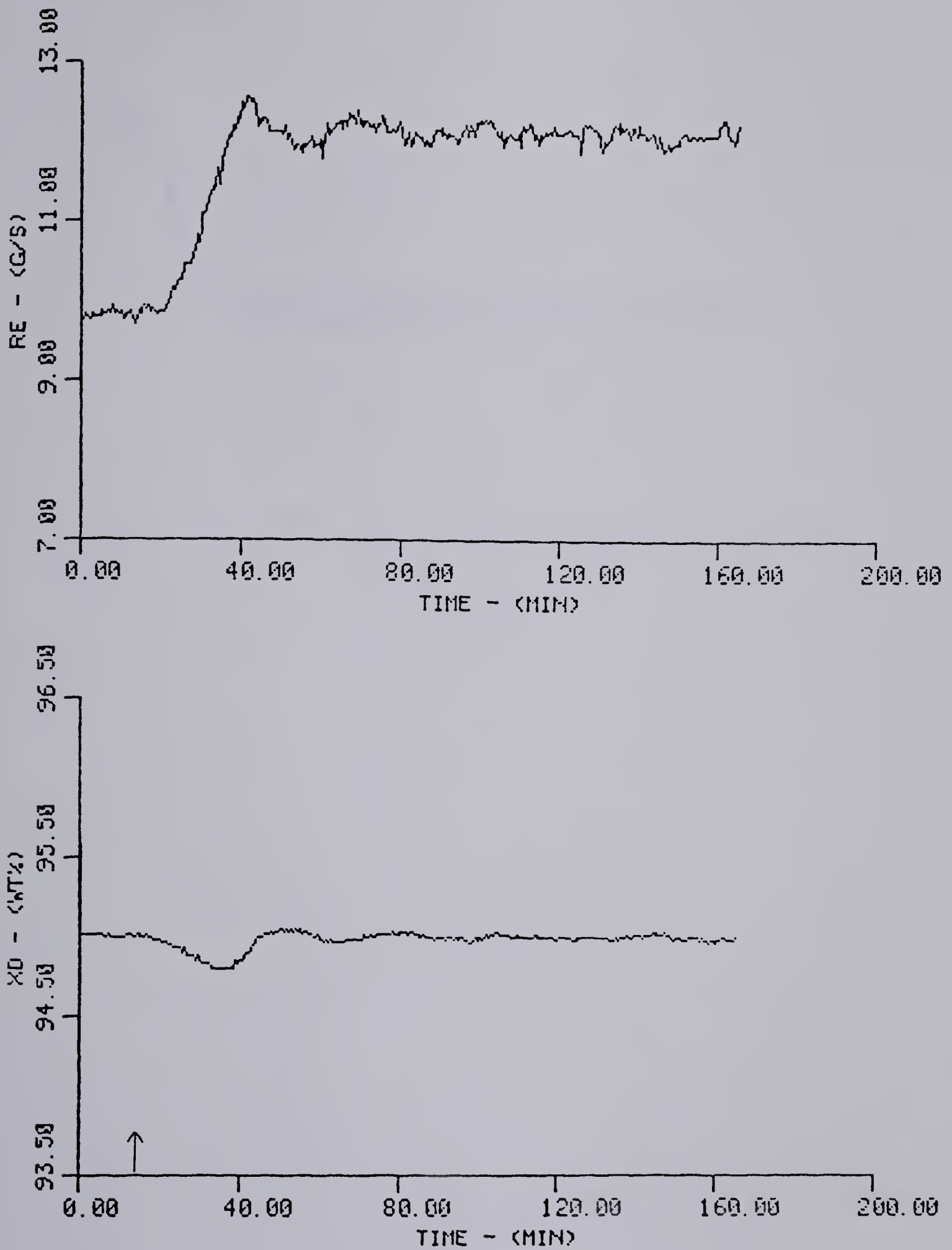


Figure 5.3 PID Control of Top Composition for a -25% Step Change in Feed Flow Rate

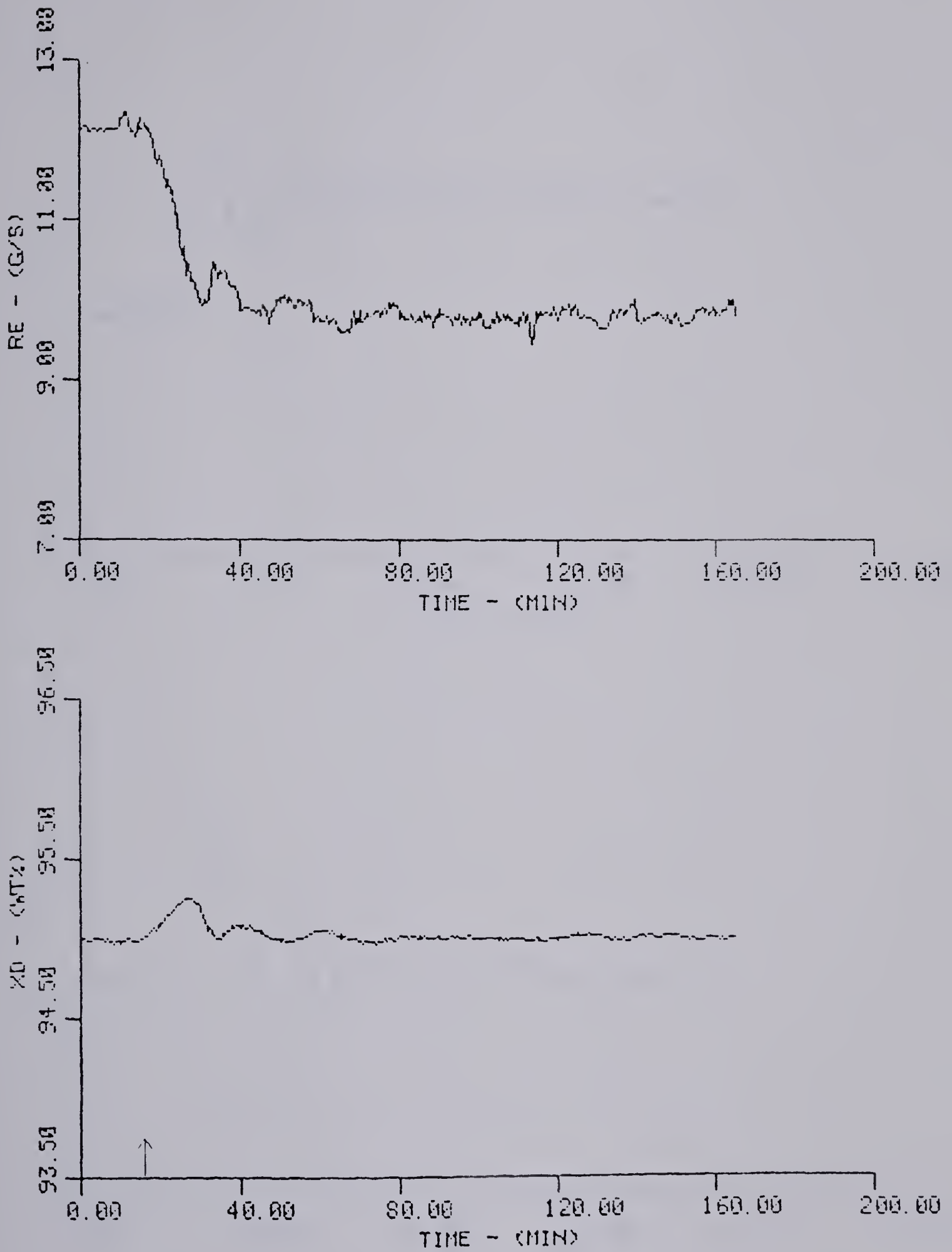


Figure 5.4 PID Control of Top Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value

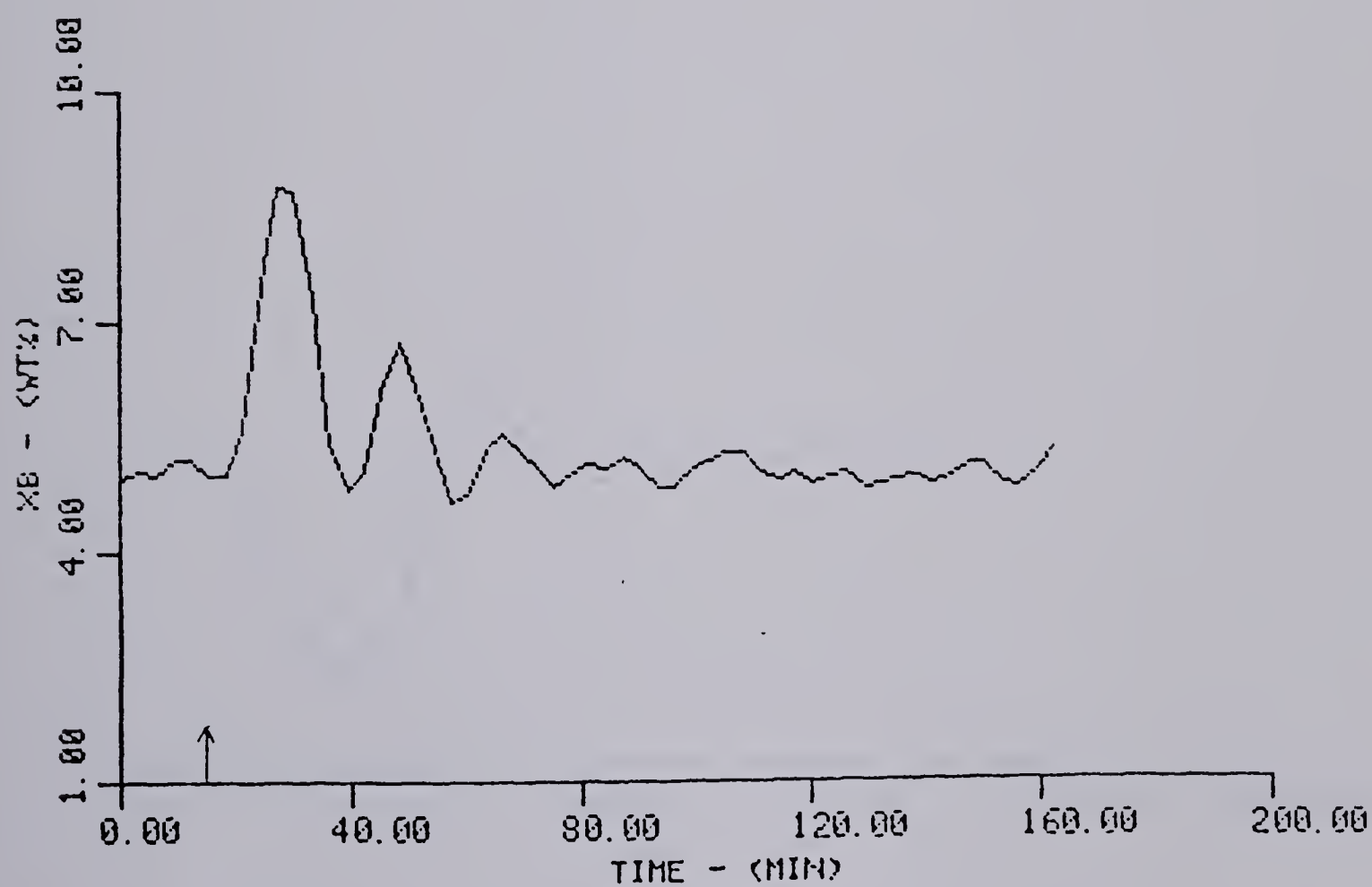
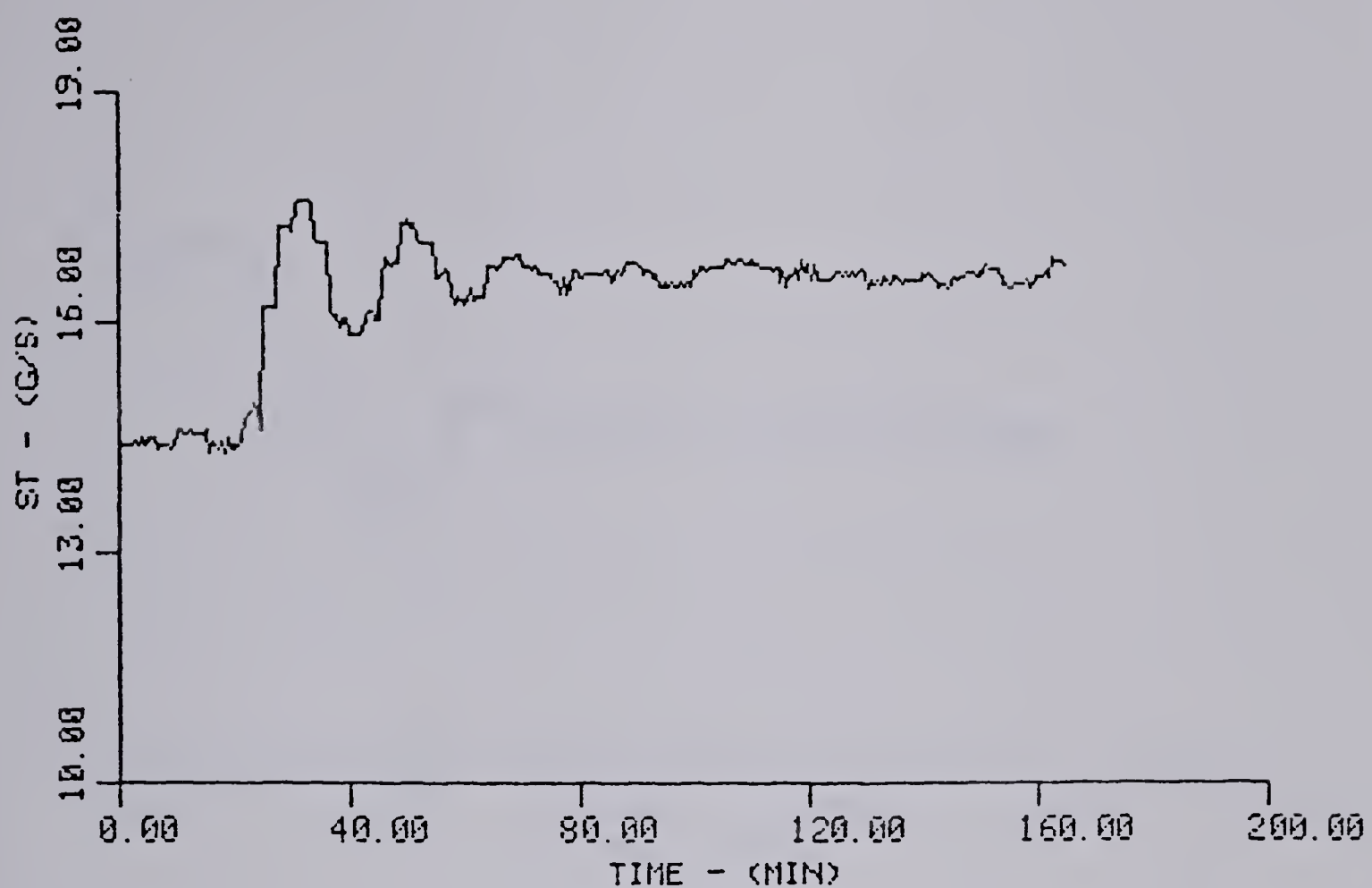


Figure 5.5 PID Control of Bottom Composition for a +25% Step Change in Feed Flow Rate

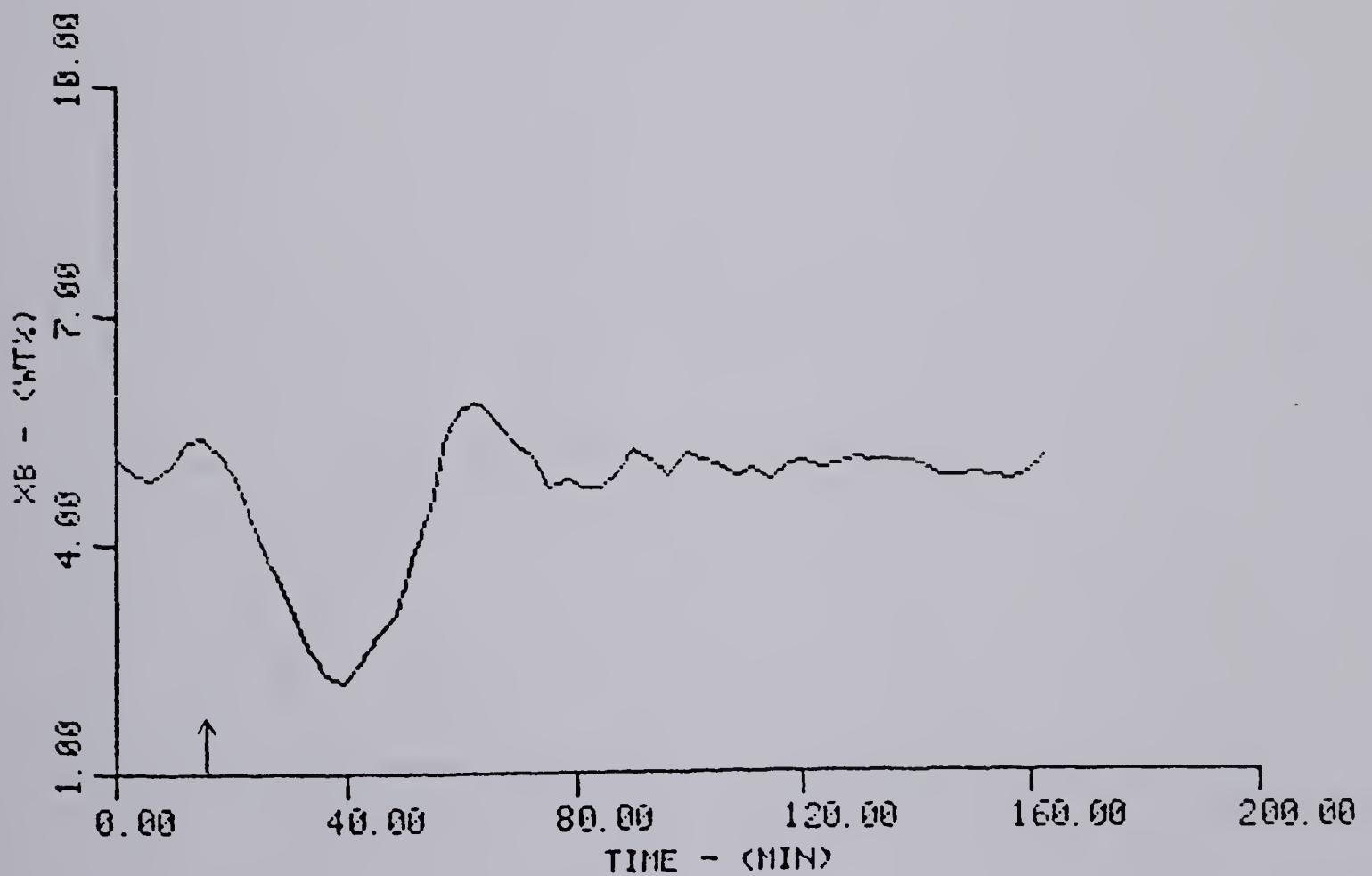
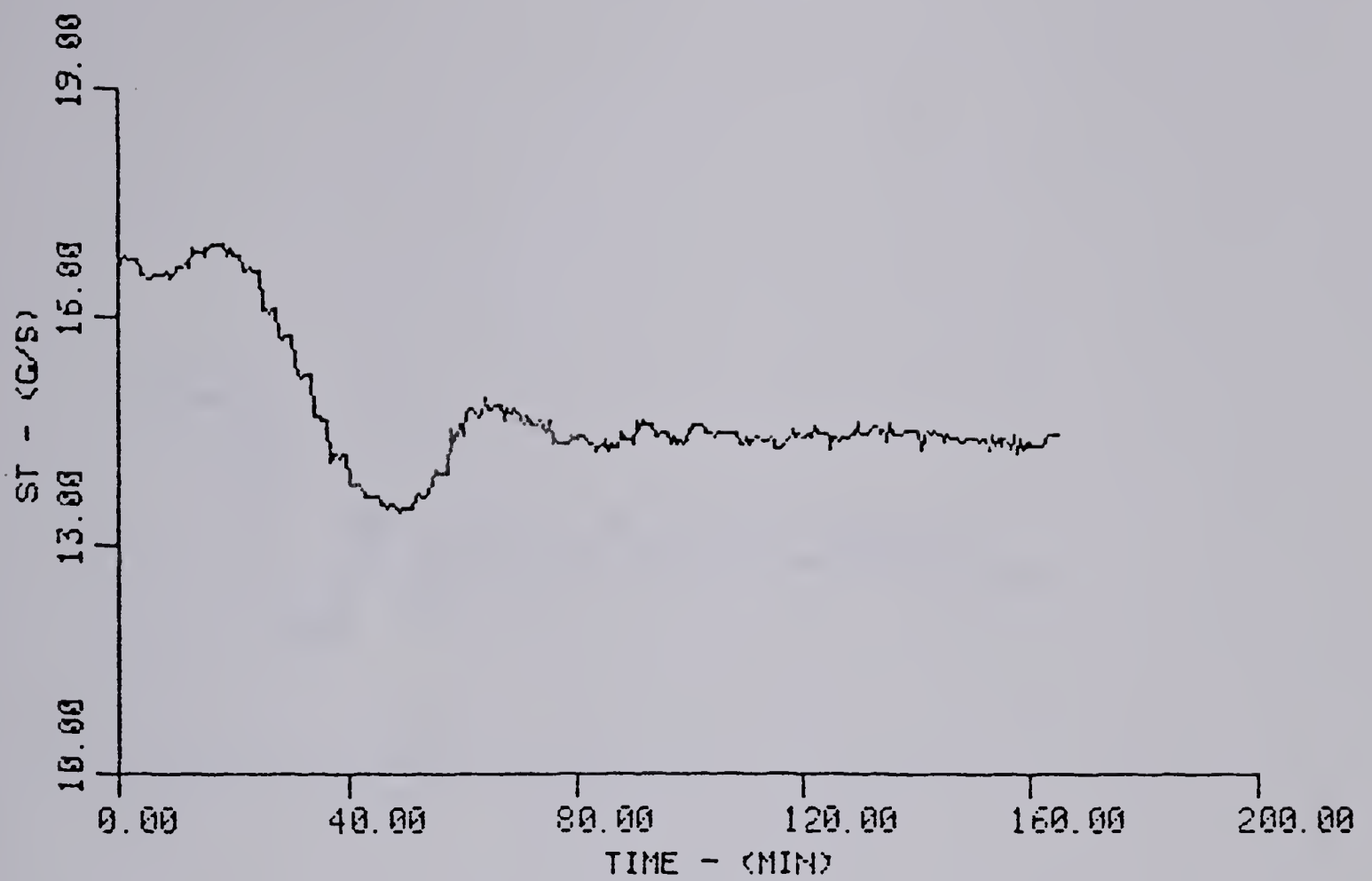


Figure 5.6 PID Control of Bottom Composition for a Step Decrease in Feed Flow Rate to its Normal Steady State Value

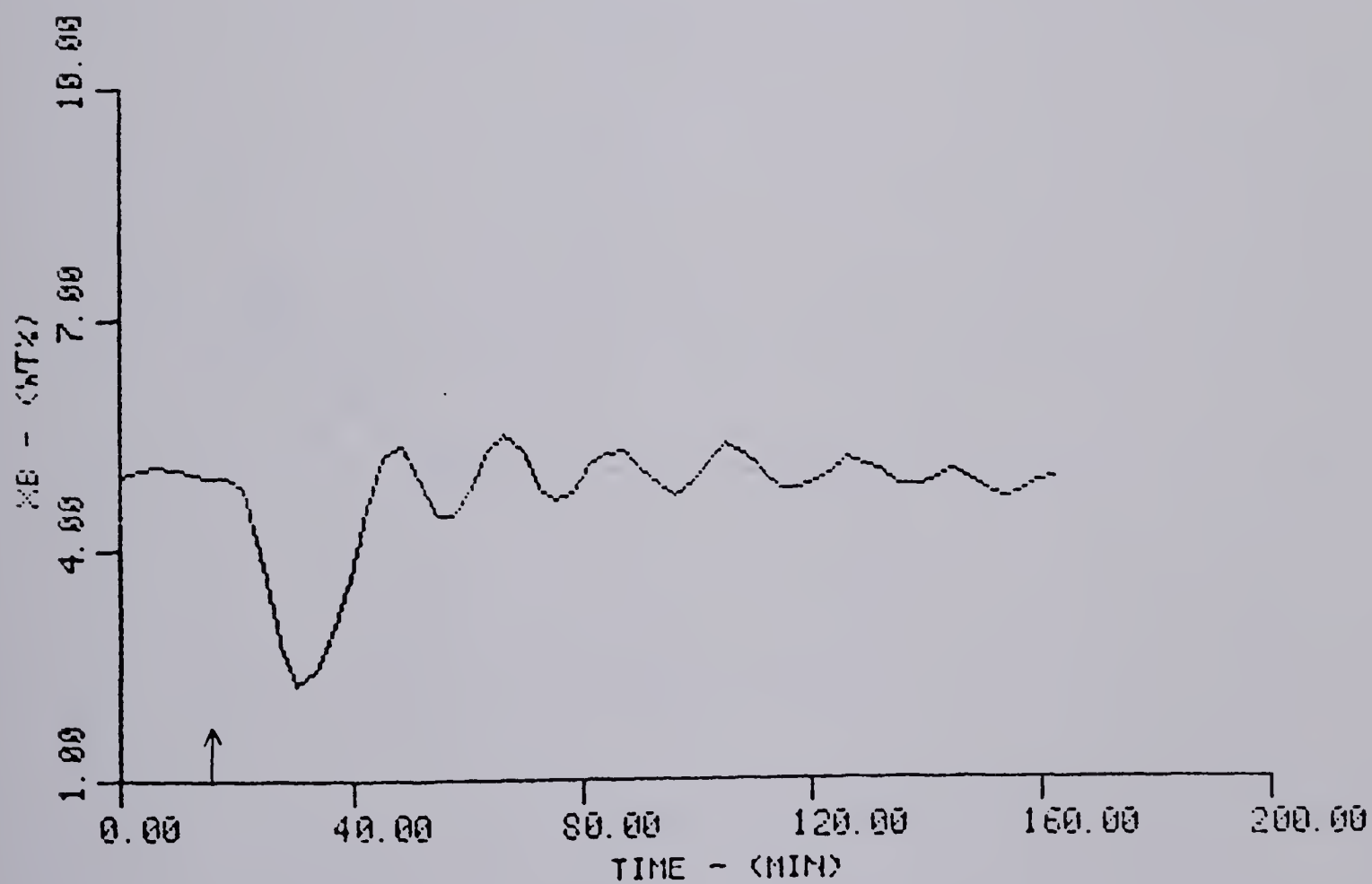
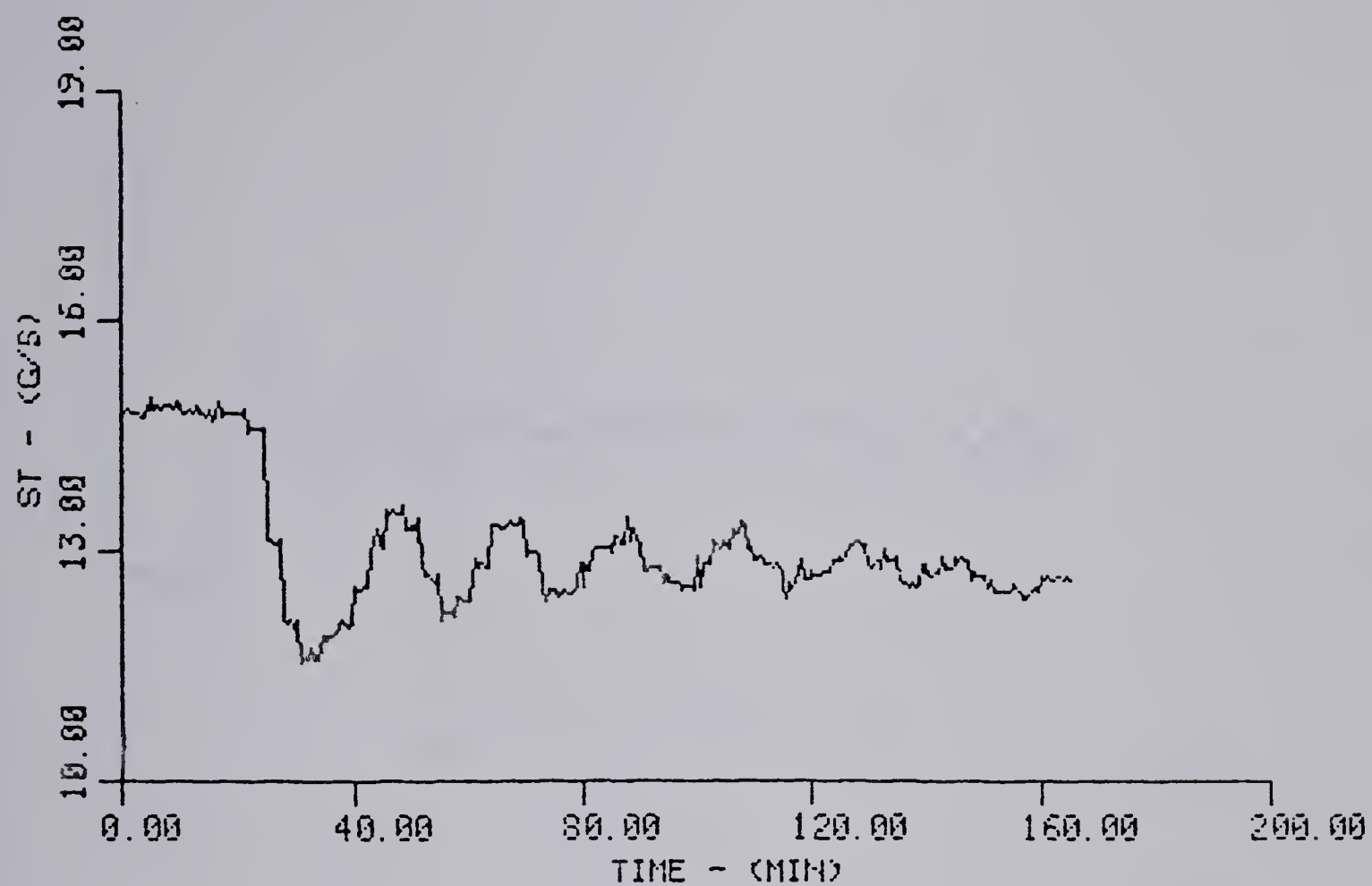


Figure 5.7 PID Control of Bottom Composition for a -25% Step Change in Feed Flow Rate

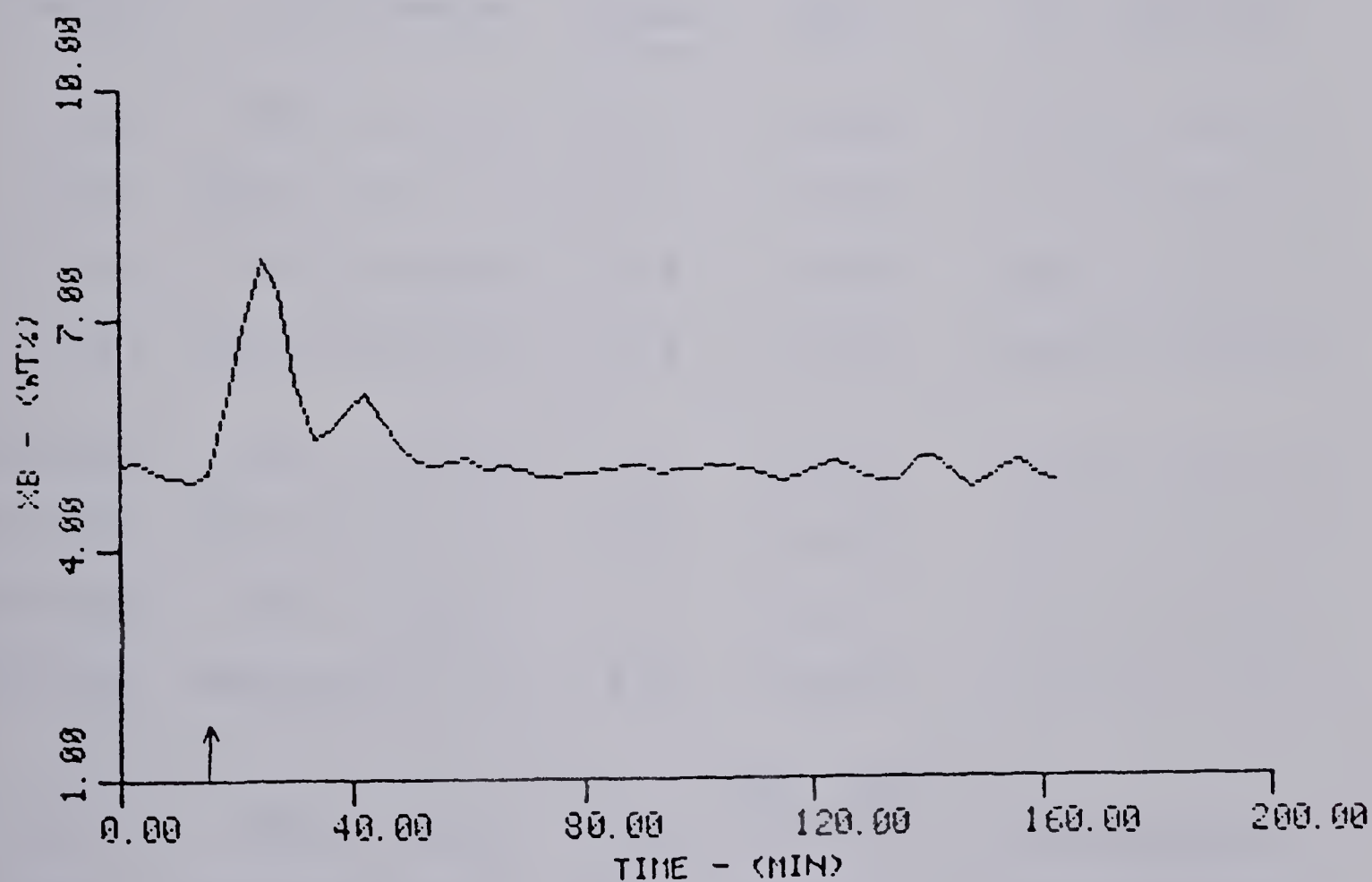
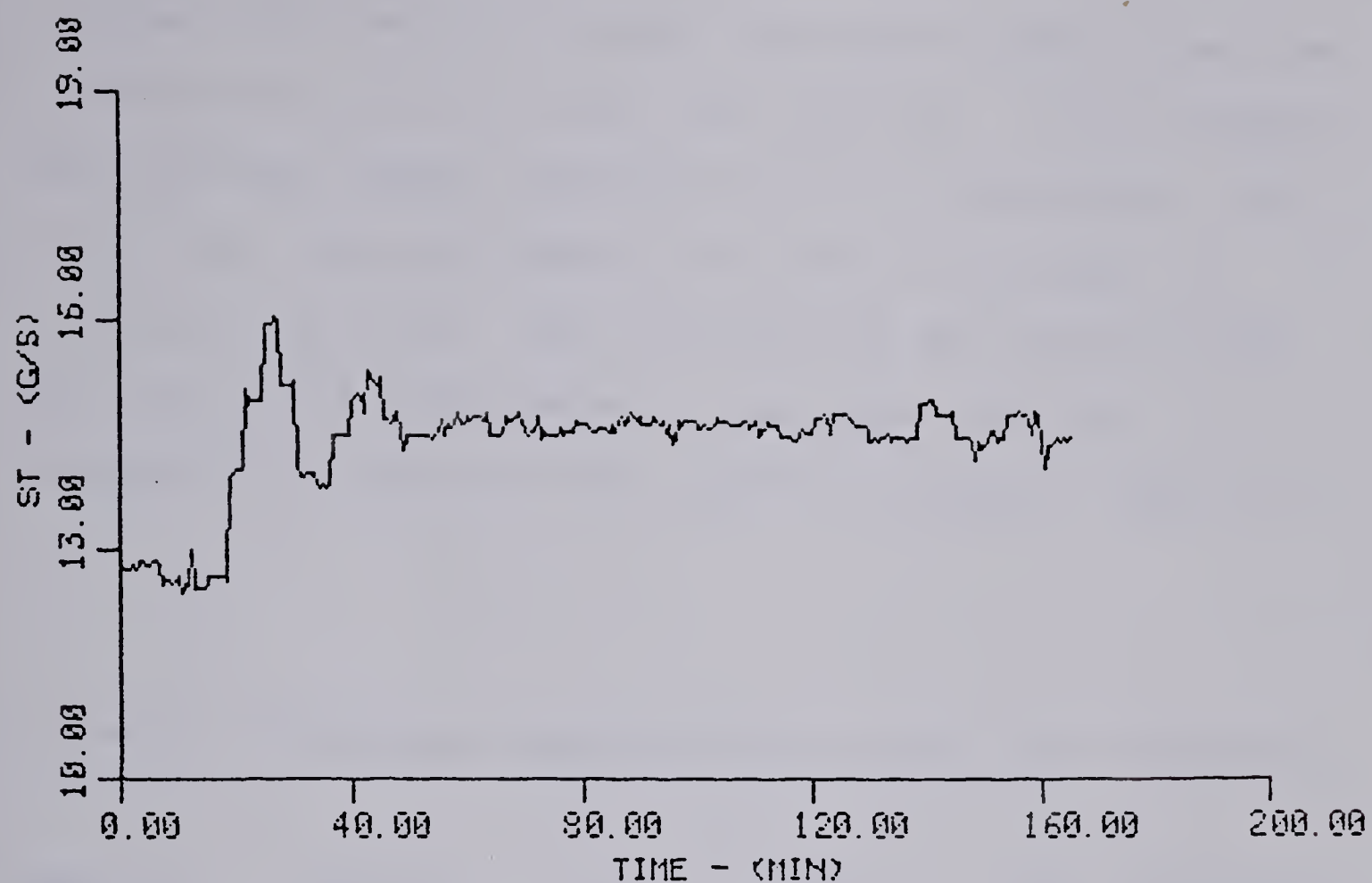


Figure 5.8 PID Control of Bottom Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value

different for the positive and the negative feed flow rate disturbances as can be seen from Table 5.1. This is due to the nonlinear dynamic characteristics of the column. The results for the dual composition control are shown in Figures 5.9 to 5.12. The PID constants are tuned on the basis of a -25% step change in feed flow rate only. A summary of the SAE for these results is given in Table 5.1.

Table 5.1 Summary of Absolute Error Values For PID Control

Type of Control	Type of Disturbance	Figure	SAE (wt%)	PID Constants $k_p / k_i / k_d$
Top	+25% to feed	5.1	0.460	2.81/0.73/0.65
Top	Return to S.S.	5.2	0.785	2.81/0.73/0.65
Top	-25% to feed	5.3	3.870	2.81/0.73/0.65
Top	Return to S.S.	5.4	3.410	2.81/0.73/0.65
Bottom	+25% to feed	5.5	20.72	-.50/-.13/-.08
Bottom	Return to S.S.	5.6	26.05	-.50/-.13/-.08
Bottom	-25% to feed	5.7	18.43	-.70/-.15/-.67
Bottom	Return to S.S.	5.8	14.00	-.70/-.15/-.67
Dual	+25% to feed	5.9	top / bot 2.75/27.3	top constants:
Dual	Return to S.S.	5.10	2.87/27.1	3.57/0.93/0.83
Dual	-25% to feed	5.11	2.13/21.7	bot constants:
Dual	Return to S.S.	5.12	2.30/23.0	-.58/-.13/-.13

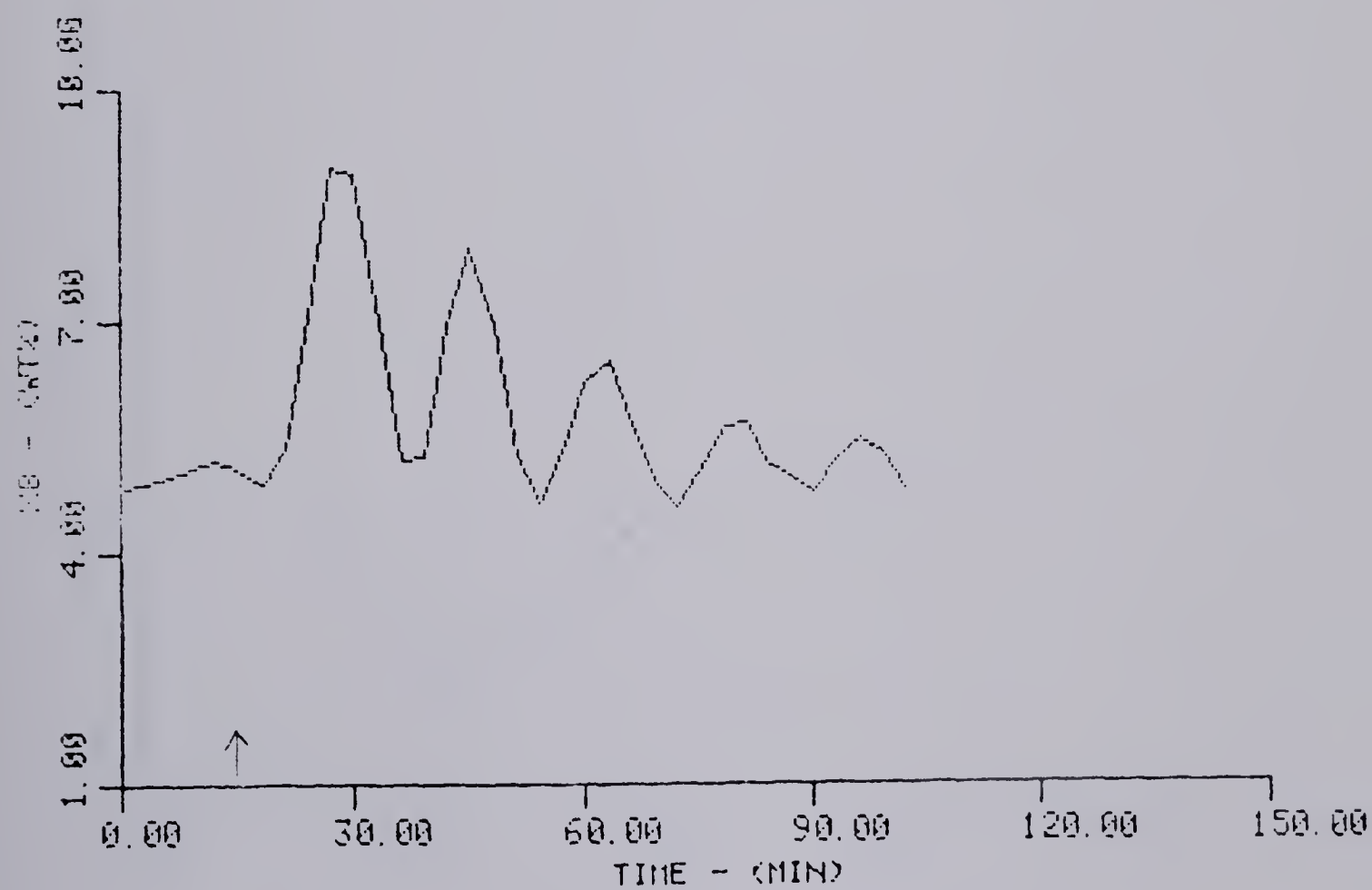
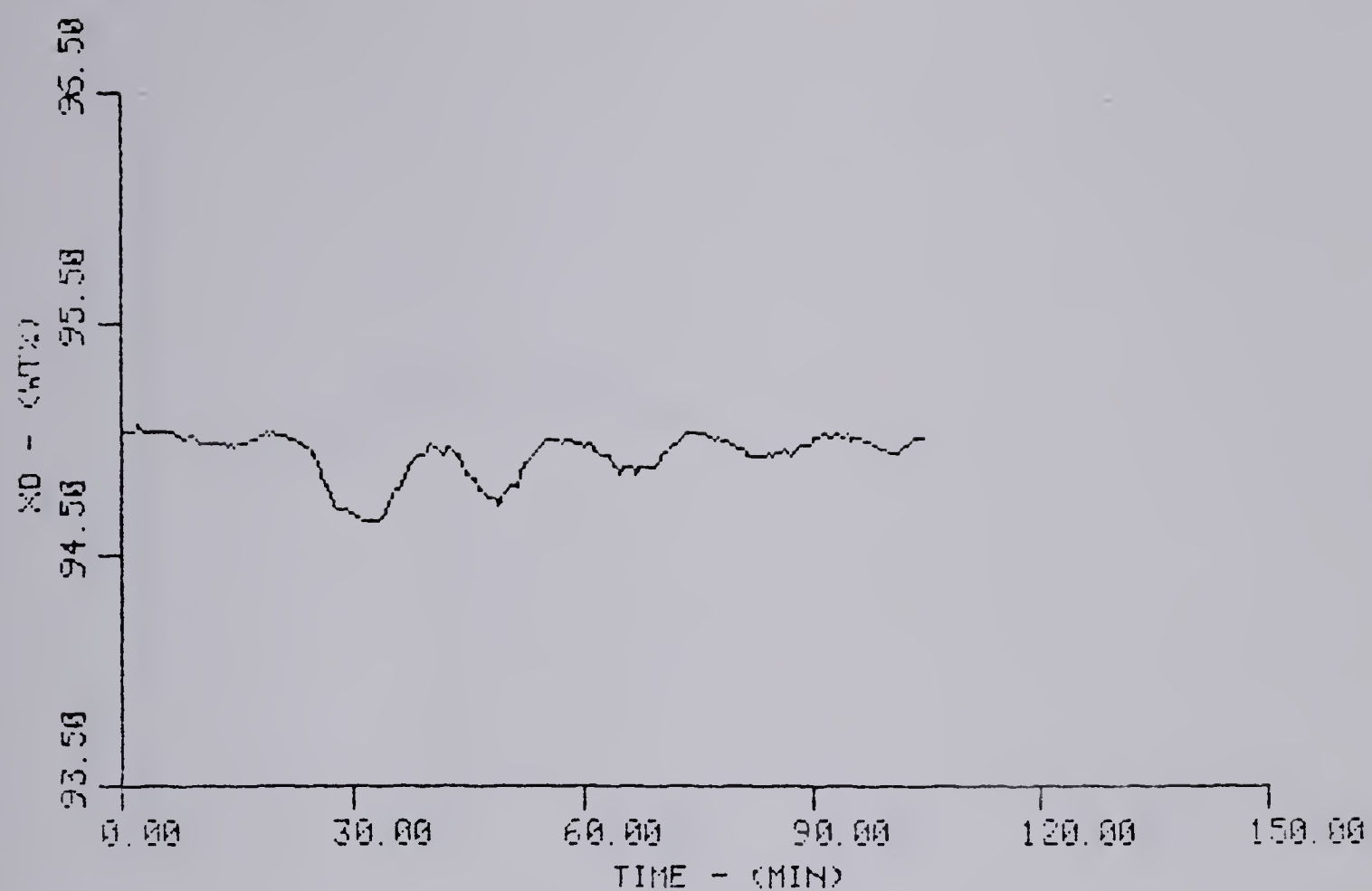


Figure 5.9 Multiloop PID Control of Terminal Compositions for a +25% Step Change in Feed Flow Rate

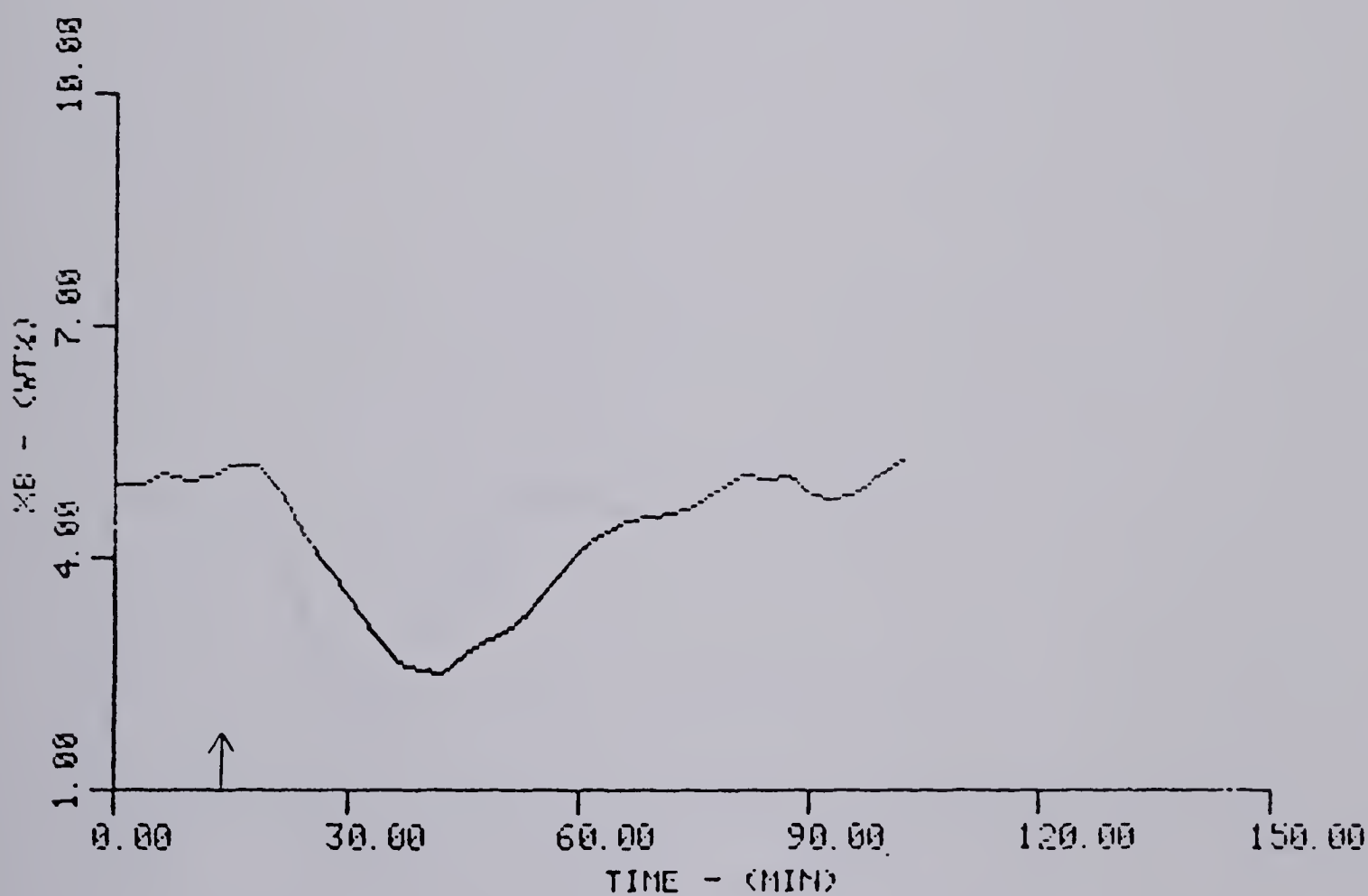
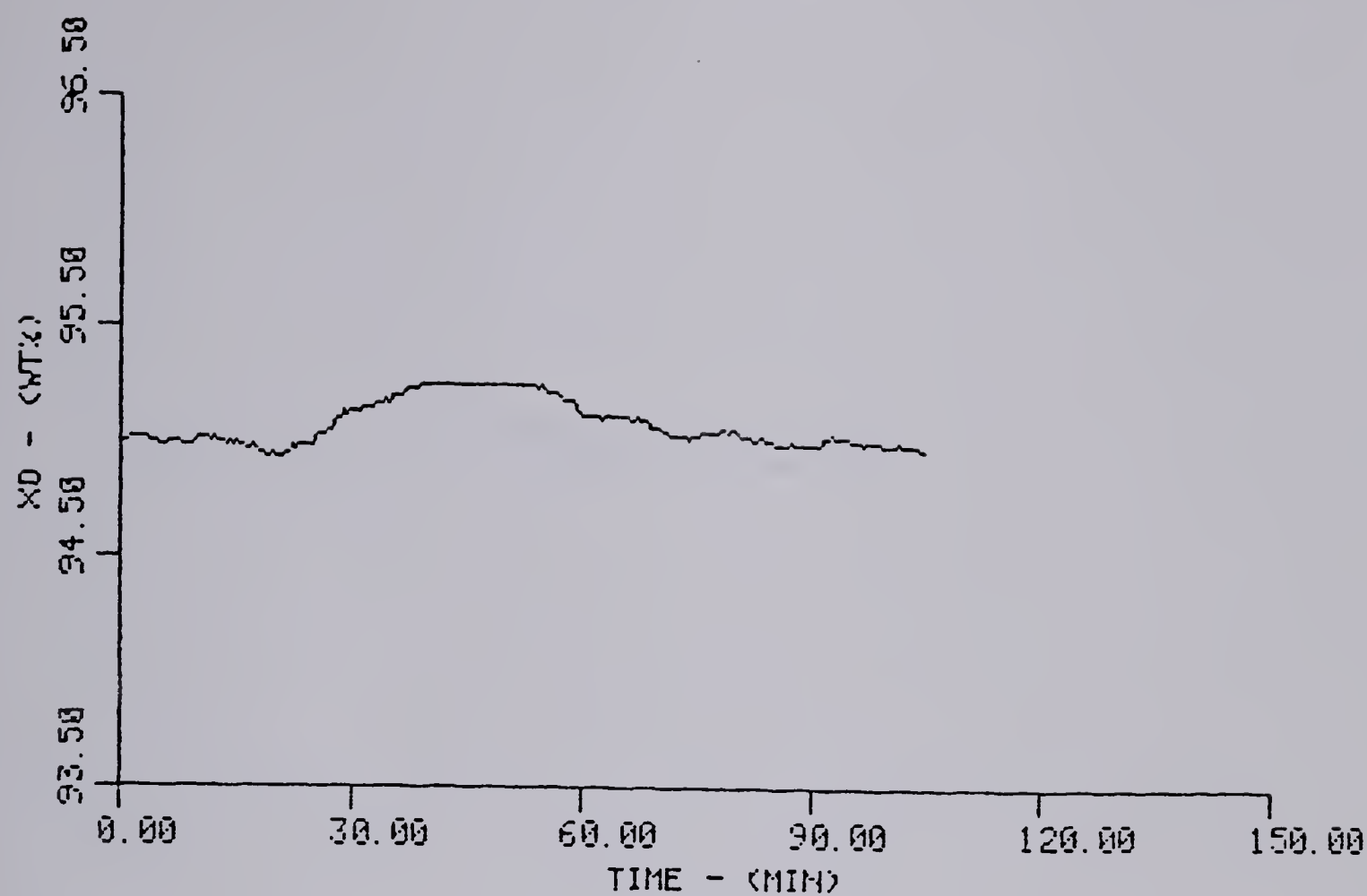


Figure 5.10 Multiloop PID Control of Terminal Compositions for a Step Decrease in Feed Flow Rate to its Normal Steady State Value

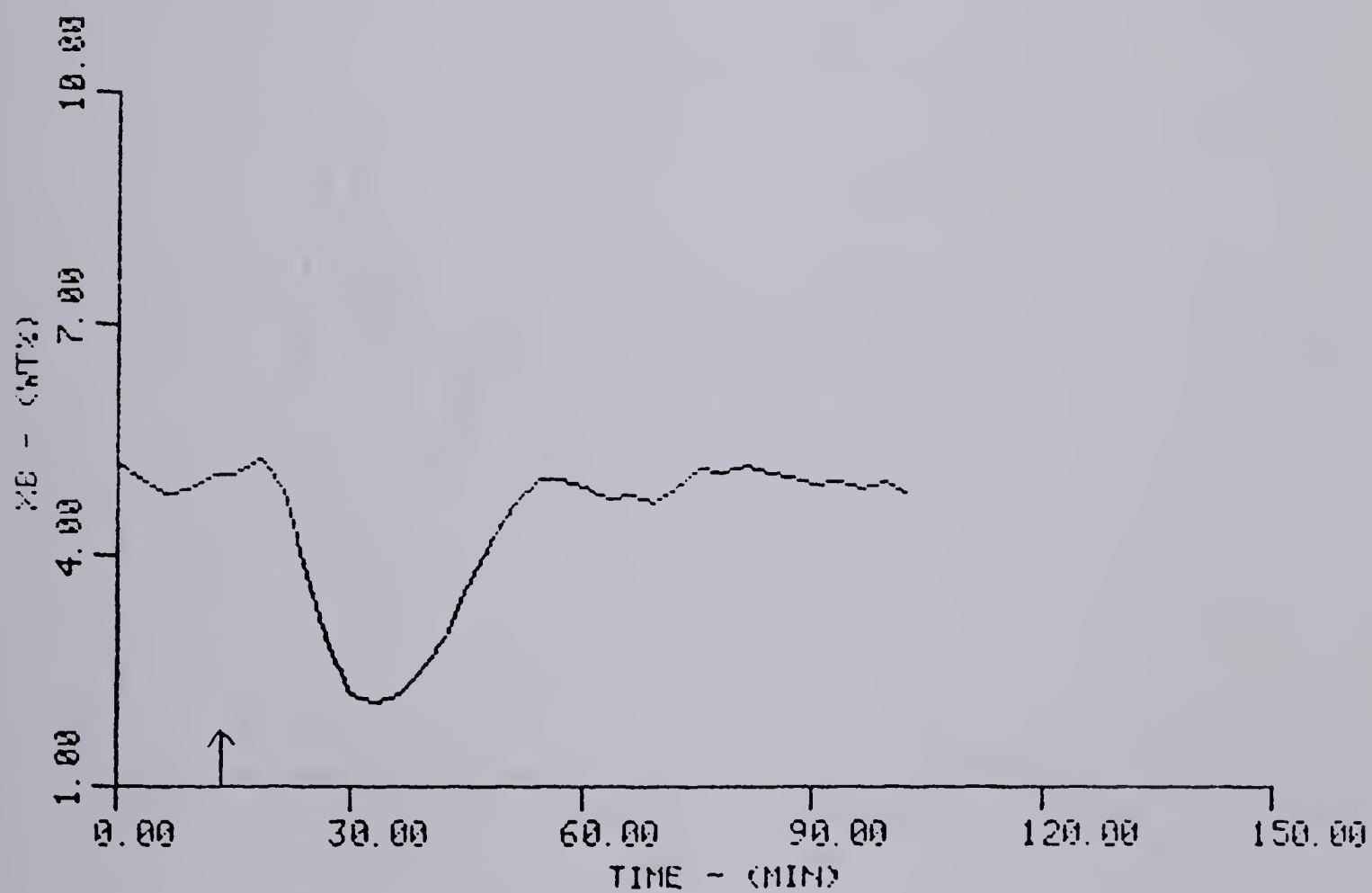
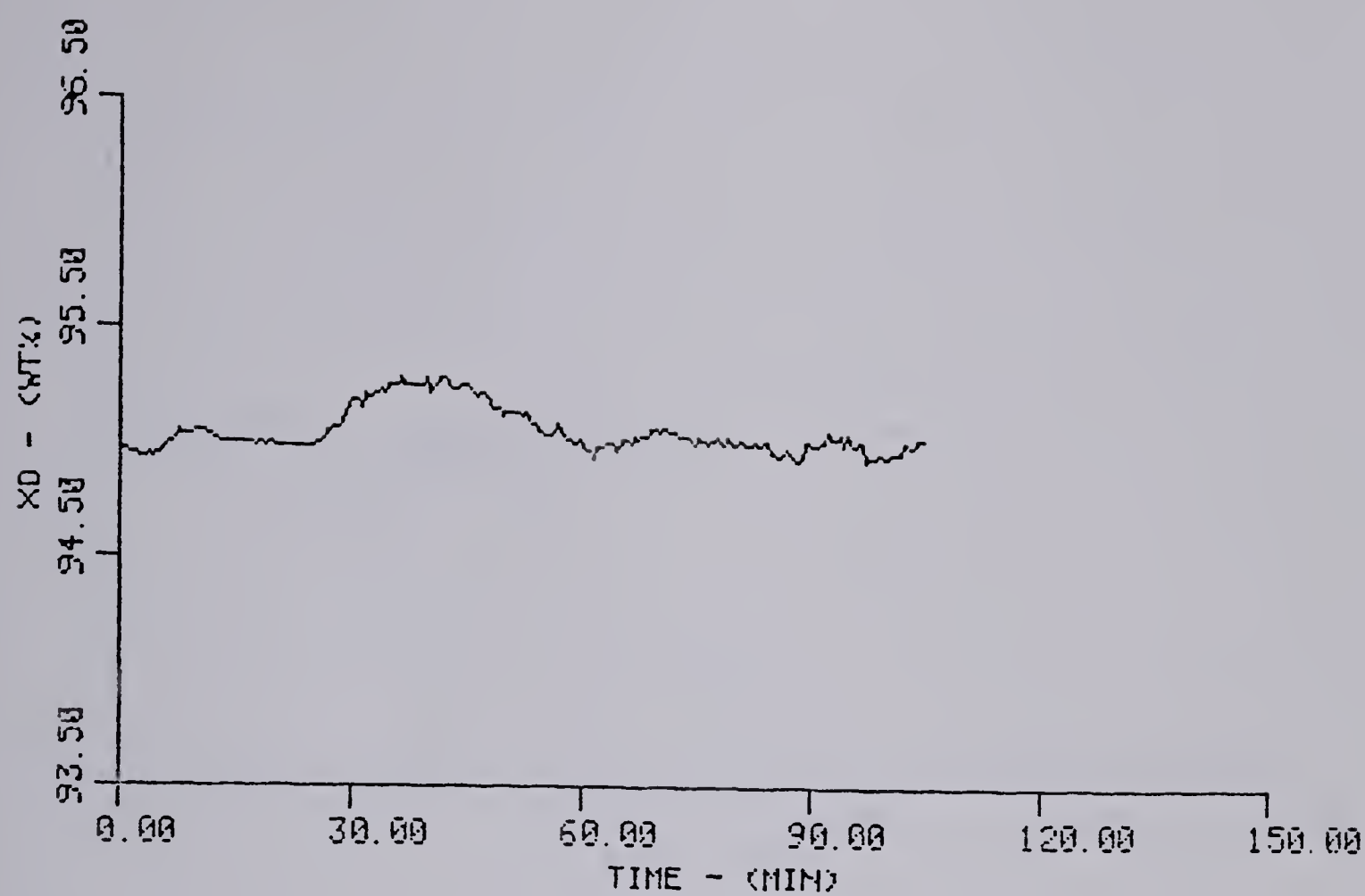


Figure 5.11 Multiloop PID Control of Terminal Compositions for a -25% Step Change in Feed Flow Rate

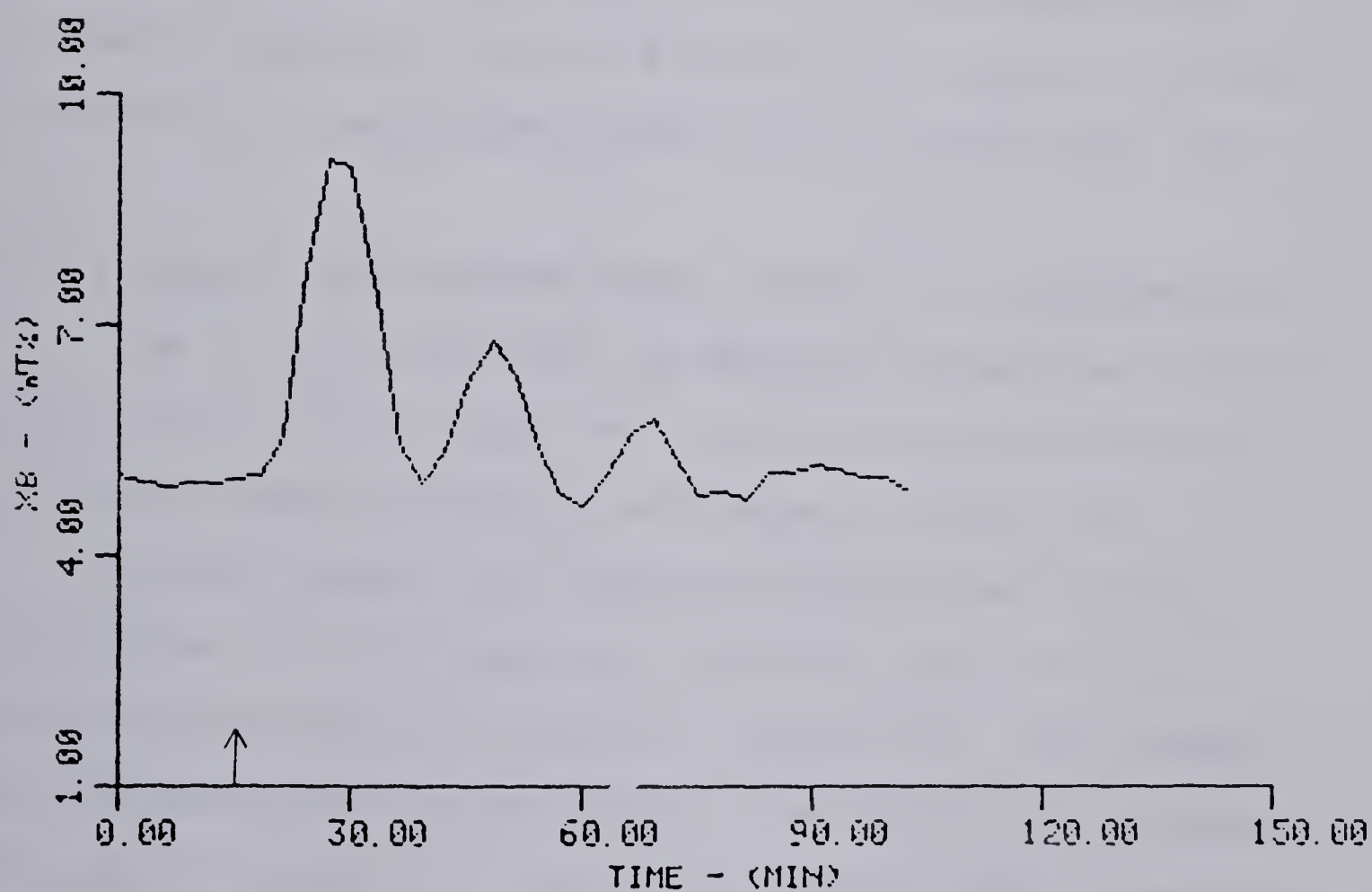
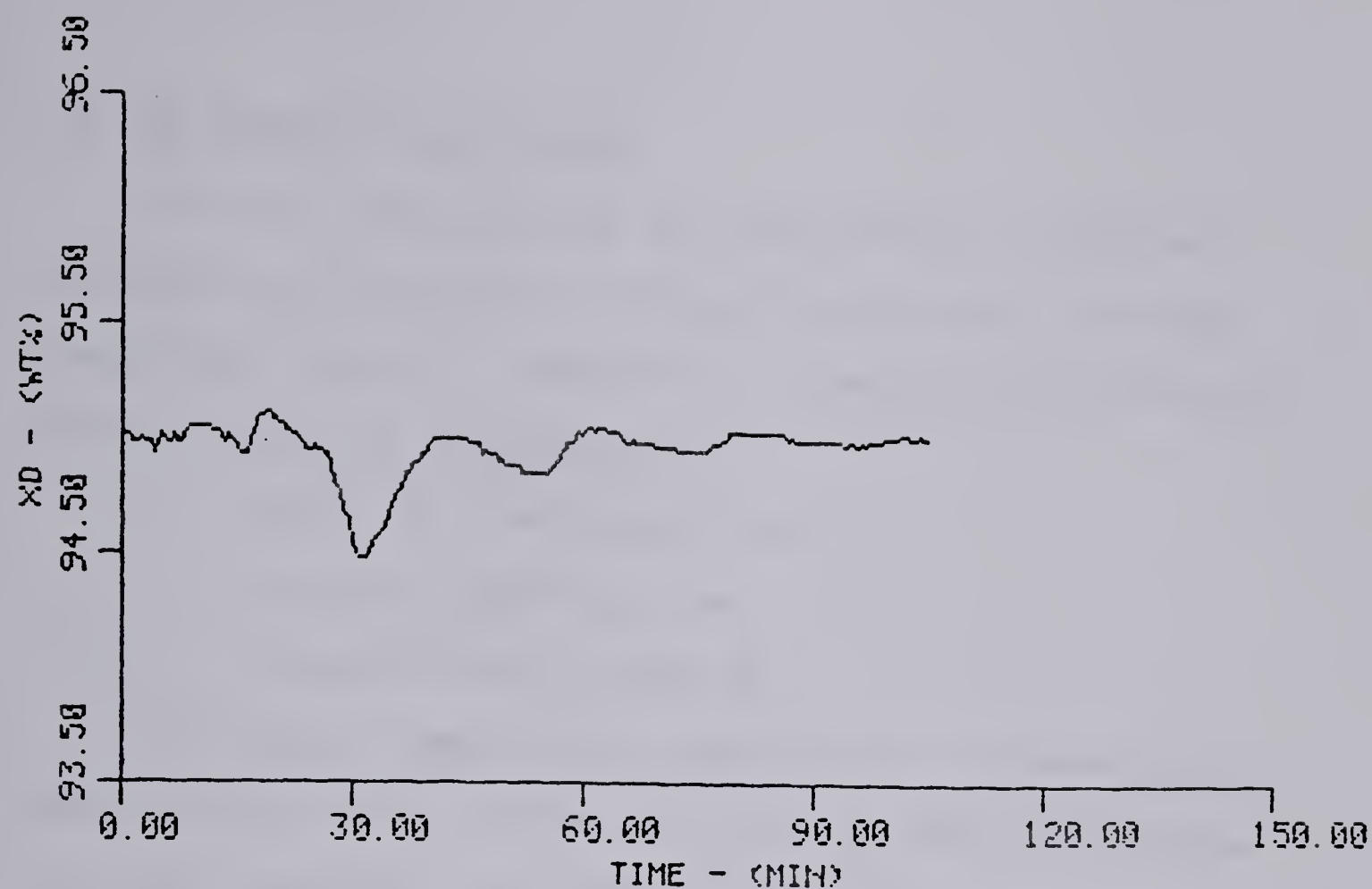


Figure 5.12 Multiloop PID Control of Terminal Compositions for a Step Increase in Feed Flow Rate to its Normal Steady State Value

5.5 TOP COMPOSITION CONTROL

This section contains the experimental responses achieved using the self-tuning controller for overhead composition control. Results are presented to show the effect on control behaviour of:

- a. removing Q-weighting (Q-WT);
- b. changing sampling time;
- c. stopping identification.

Top composition control was selected to demonstrate some characteristic of the self-tuning controller because of the loop's smaller time constant and the allowable user choice of sampling rate for the overhead composition from the continuous measurement signal of the capacitance cell.

5.5.1 Control Performance Using the STC with Q-weighting

The initial controller parameters and options used for self-tuning control when the column was subjected to the different feed disturbances are shown in Table 5.2.

The top composition response when the self-tuning controller is first made operative with zero initial parameter estimates is shown in Figure 5.13. The changes in the parameters during the initial tuning period are shown in Figures 5.14 and 5.15. It can be seen that even starting with zero initial estimates, the parameters have converged after less than 30 sampling intervals. With the parameters having converged, the feed flow rate was decreased by 25%,

Table 5.2 Controller Parameters and Options for STC Top Composition Control

Identification Scheme	: Recursive Square Root
G Polynomial	: EB
Number of Parameters	: NG1 = 3 NF = 2 NH = 1 K1 = 1
Parameter Fixed	: $h_0 = -1$
Initial Value of Parameters	: All set to zero
Q-weighting	: $k_p = 3.676 \text{ (g/s)/wt\%}$ $k_i = 0.919 \text{ (g/s)/wt\%}$
P-weighting	: Set to 1
Sampling Interval	: 1 minute
Forgetting Factor	: 0.99
Initial $\underline{\underline{P}}^C$ Matrix	: $1000\underline{\underline{I}}$
Incremental Control Limit	: 2.3 g/s

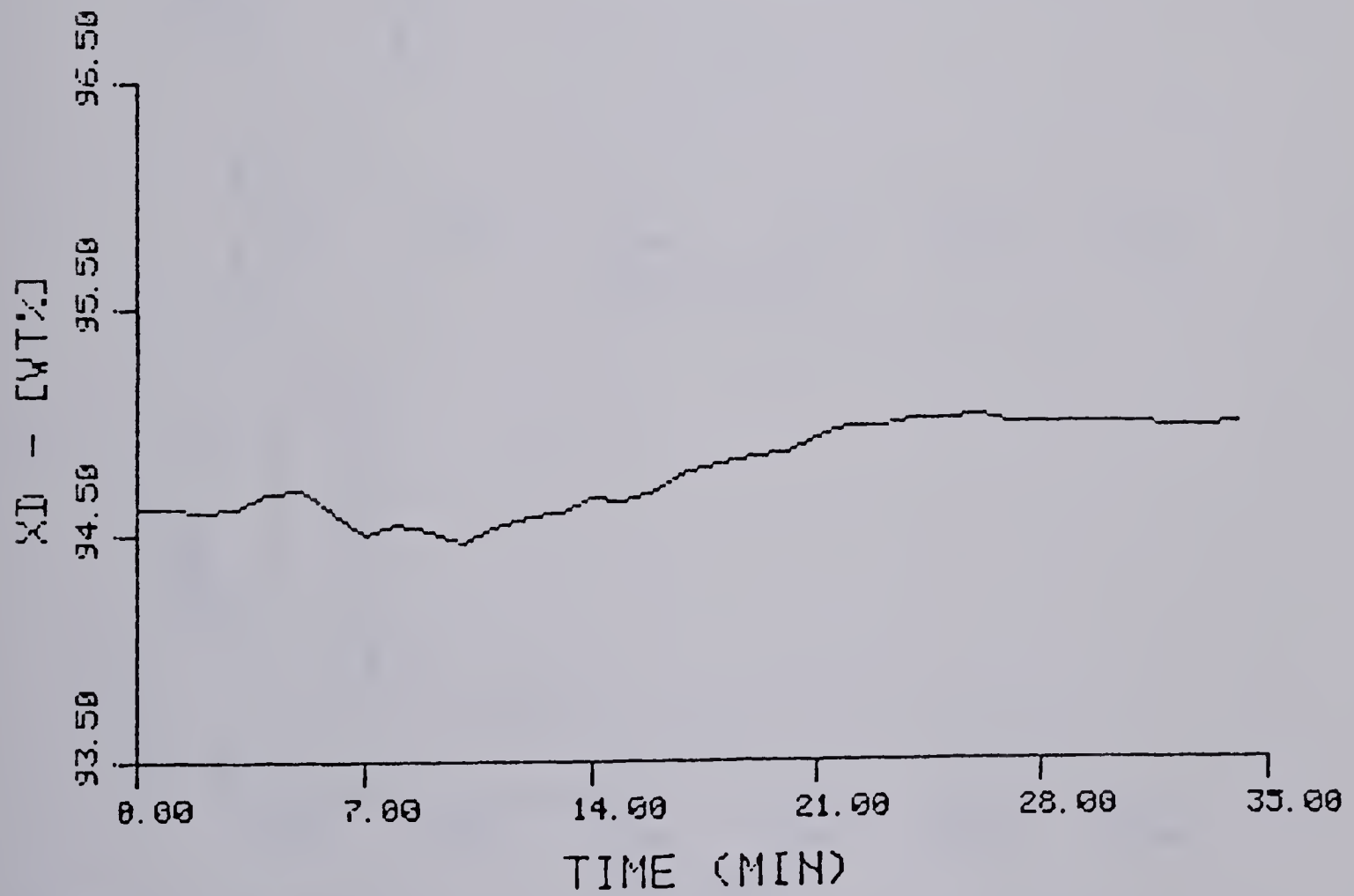
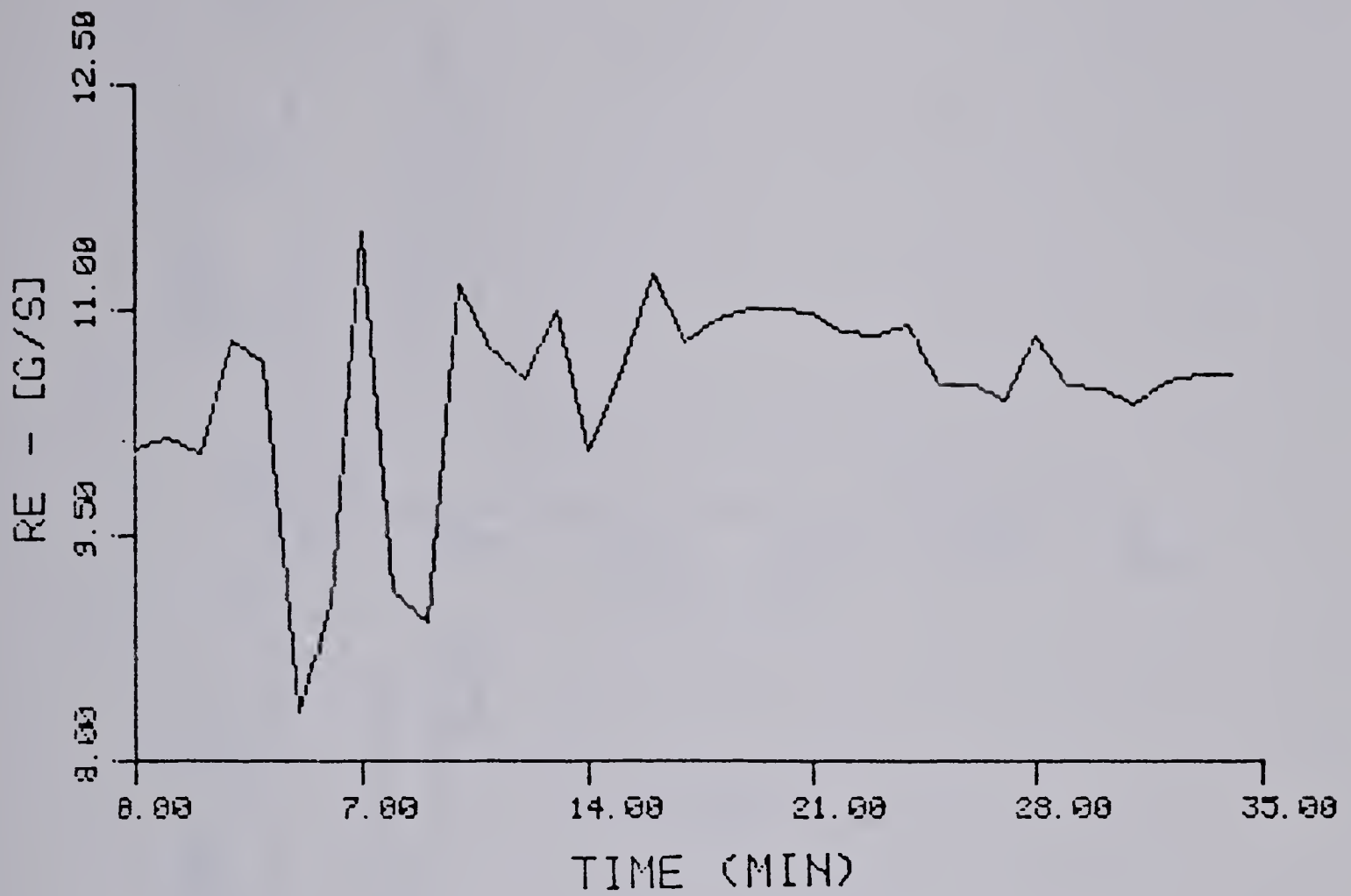


Figure 5.13 Top Composition Behaviour When Operation of the STC With Q-WT is Initiated

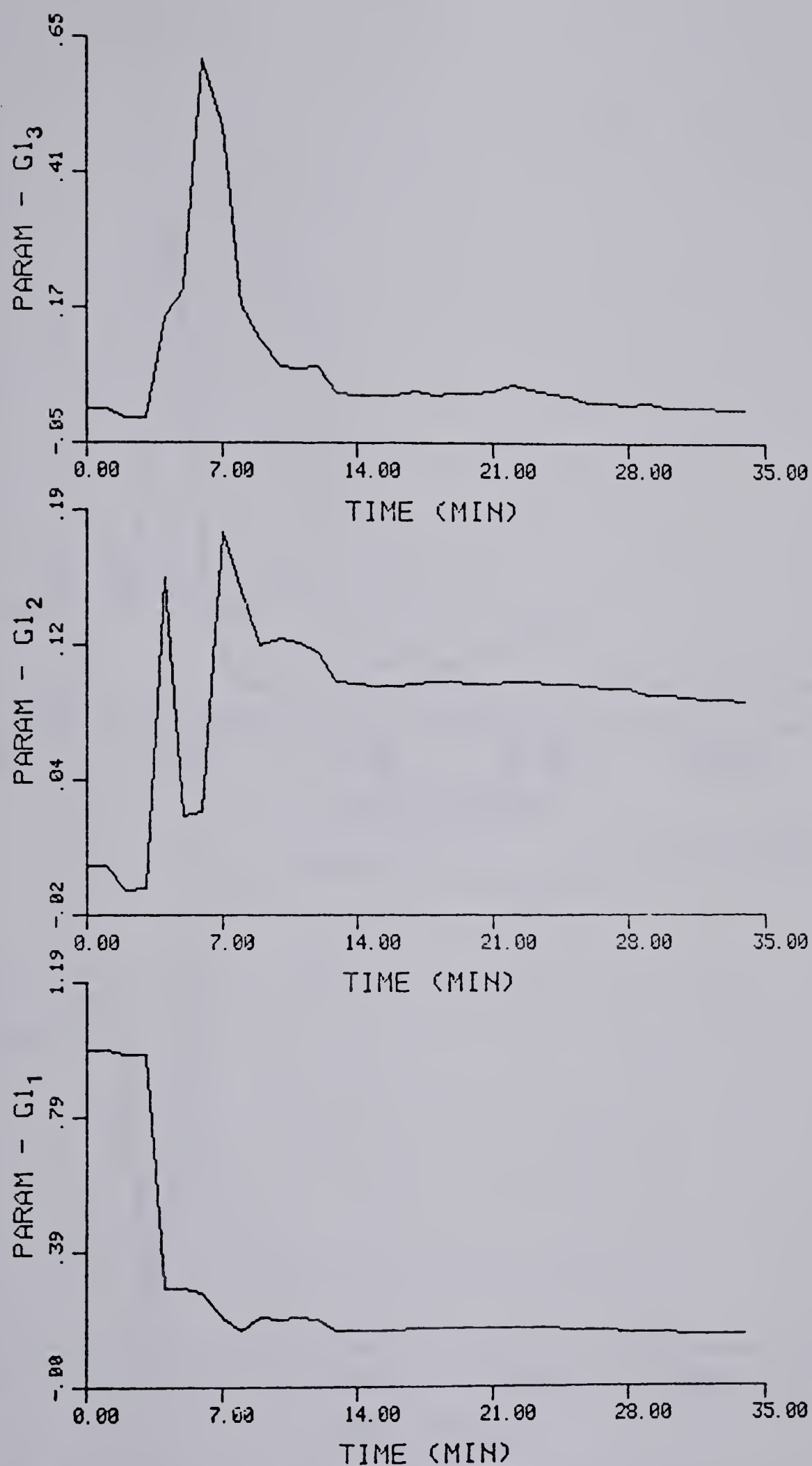


Figure 5.14 Variation of G1 Parameters in the Initial Tuning Period for STC With Q-WT

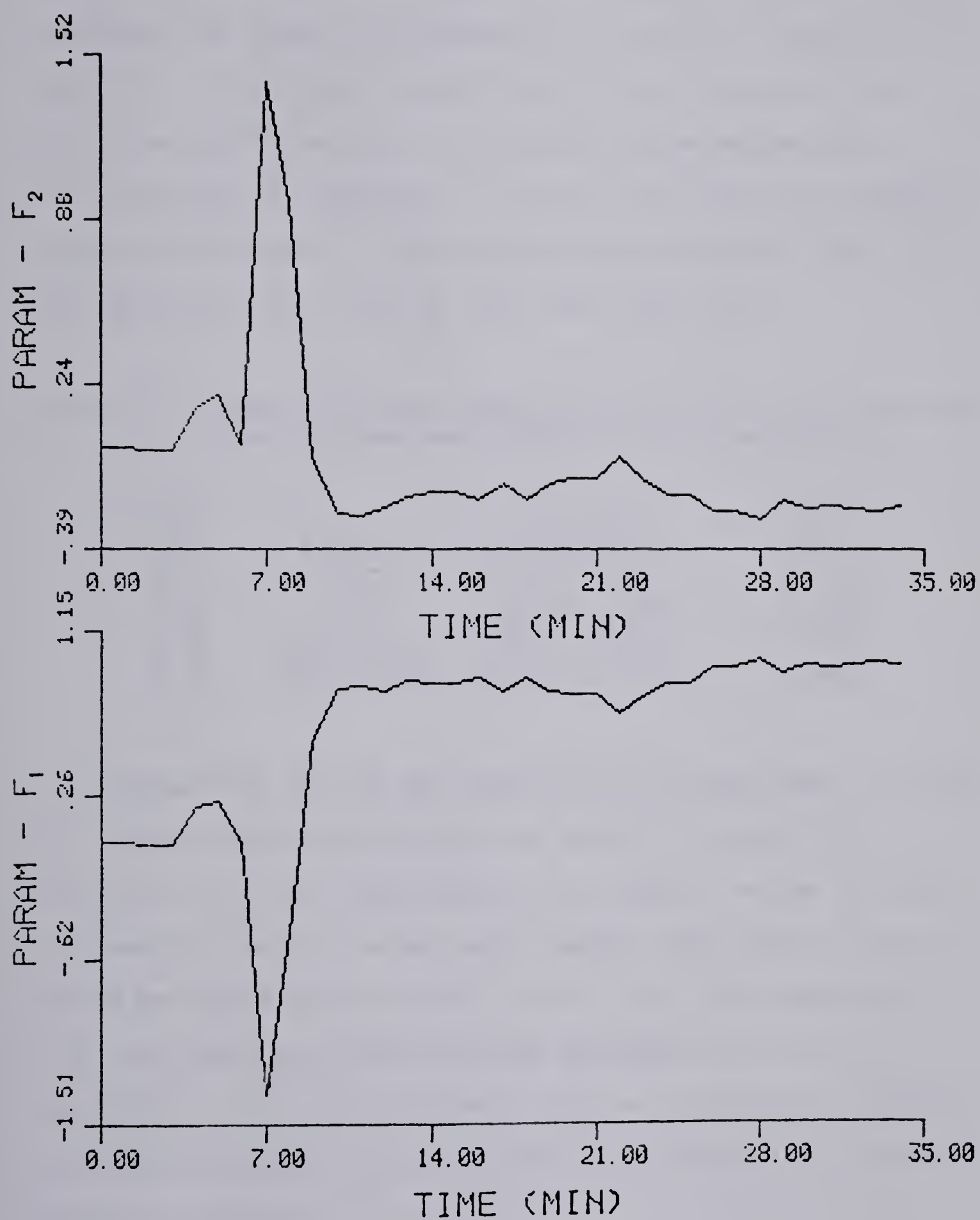


Figure 5.15 Variation of F Parameters in the Initial Tuning Period for STC With Q-WT

the system allowed to reach steady state and the flow increased to its normal steady state rate. The controlled responses for these disturbances are shown in Figures 5.16 and 5.17. To provide a quantitative indication of the controller performance, the absolute error values are calculated for 60 sampling intervals after the disturbance has been introduced. These values are given in Table 5.3. Also shown are the results from duplicate runs.

Table 5.3 Summary of Top Composition Absolute Error Values (Over 60 Samples) for STC with Q-weighting

Run No.	Figure	Type of Disturbance	SAE (wt%)
2-2	5.16	-25% to Feed	2.177
2-3	5.17	Return to S.S.	2.197
5-4	Duplicate	-25% to Feed	2.347
5-5	Duplicate	Return to S.S.	1.867

Comparison of the SAE results with those shown in Table 5.1, shows that the STC performs better than the PID controller for the feed disturbances used. While the SAE achieved by the STC is not much better than the PID case, it should be noted that the PID results were obtained only after considerable time had been expended to tune the controller. This is in marked contrast to the STC results obtained using the first Q-weighting selected (cf. values given in Table 5.2).

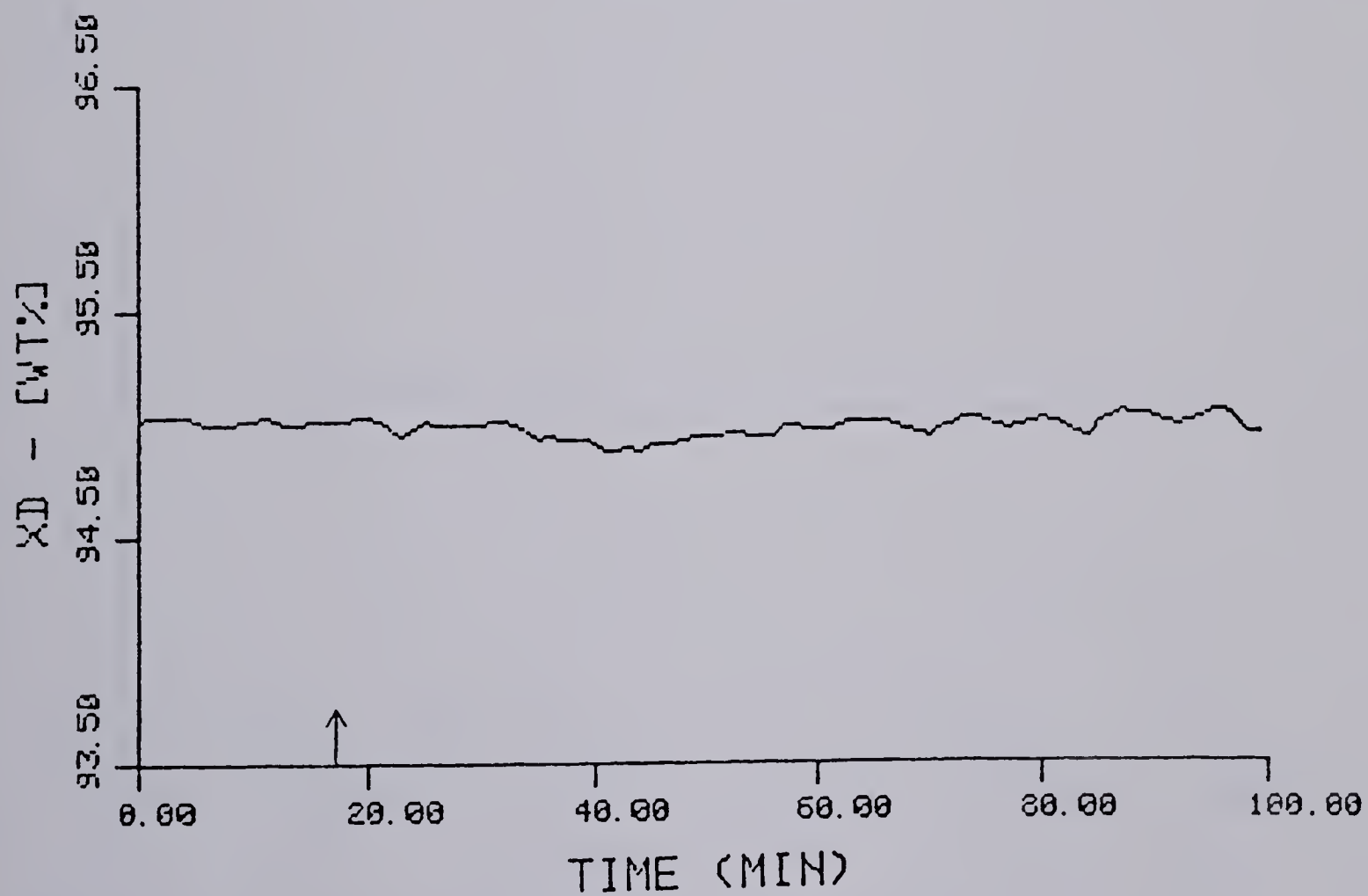
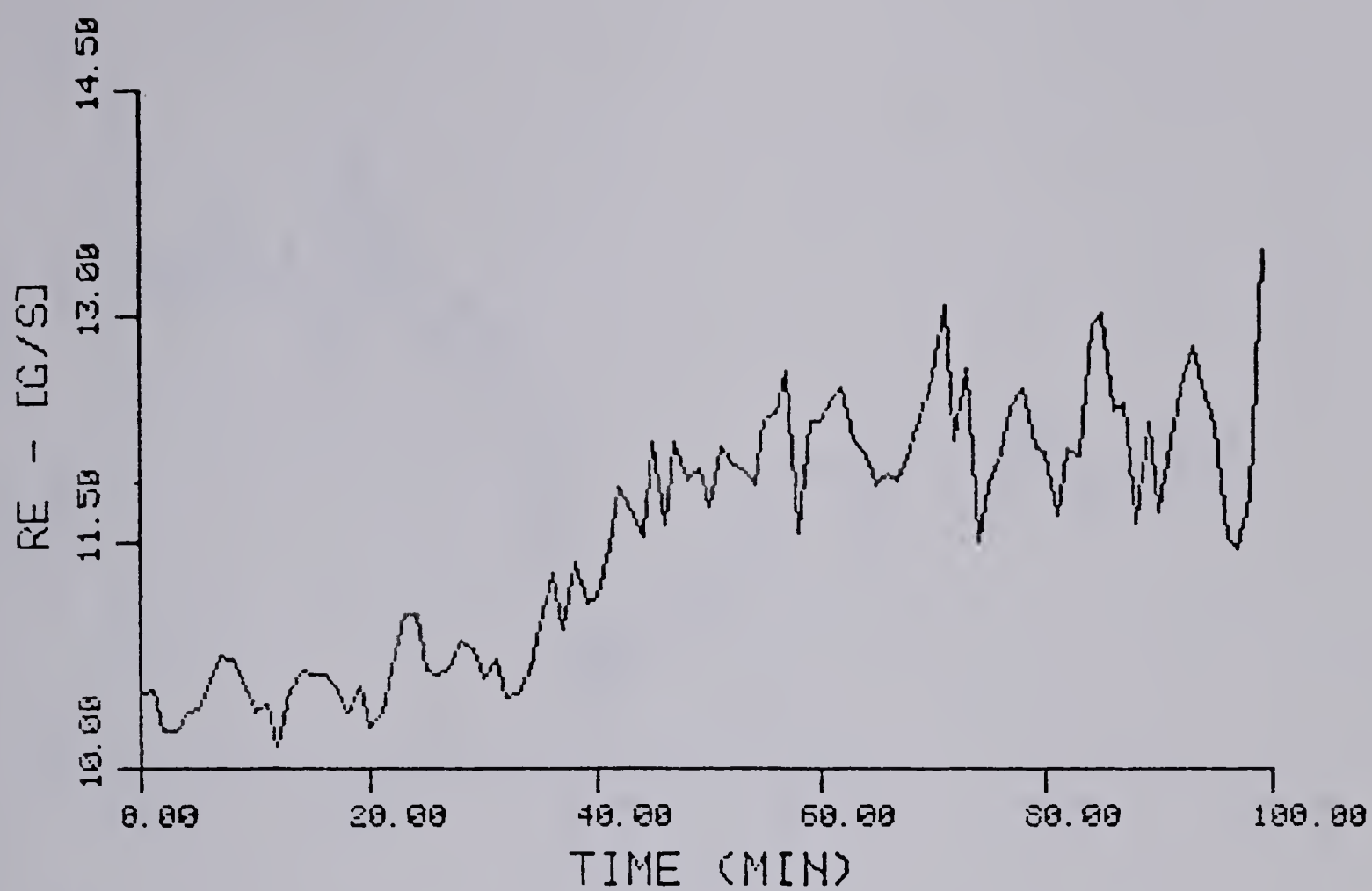


Figure 5.16 ST Control of Top Composition for a -25% Step Decrease in Feed Flow Rate With Q-WT

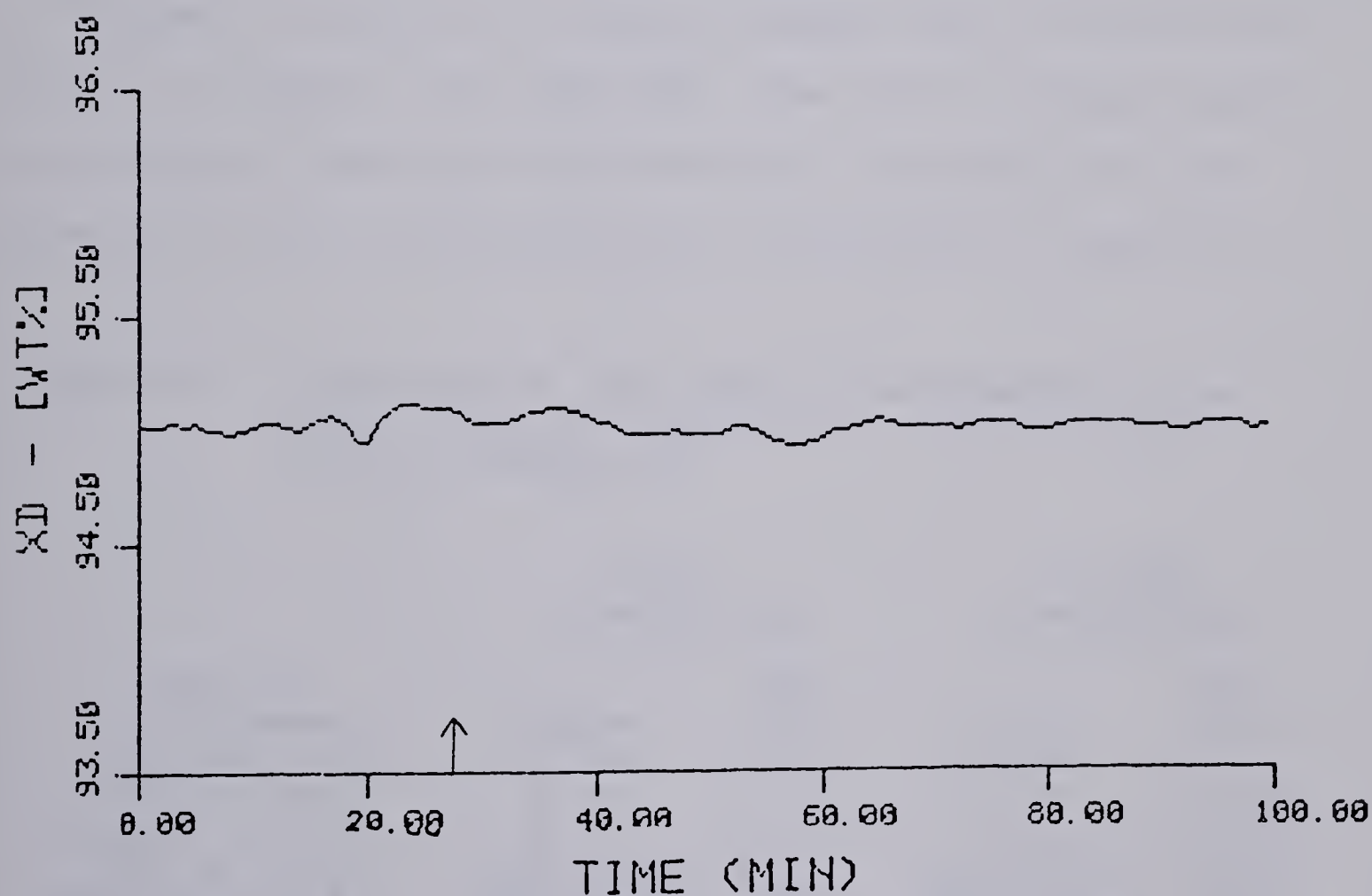
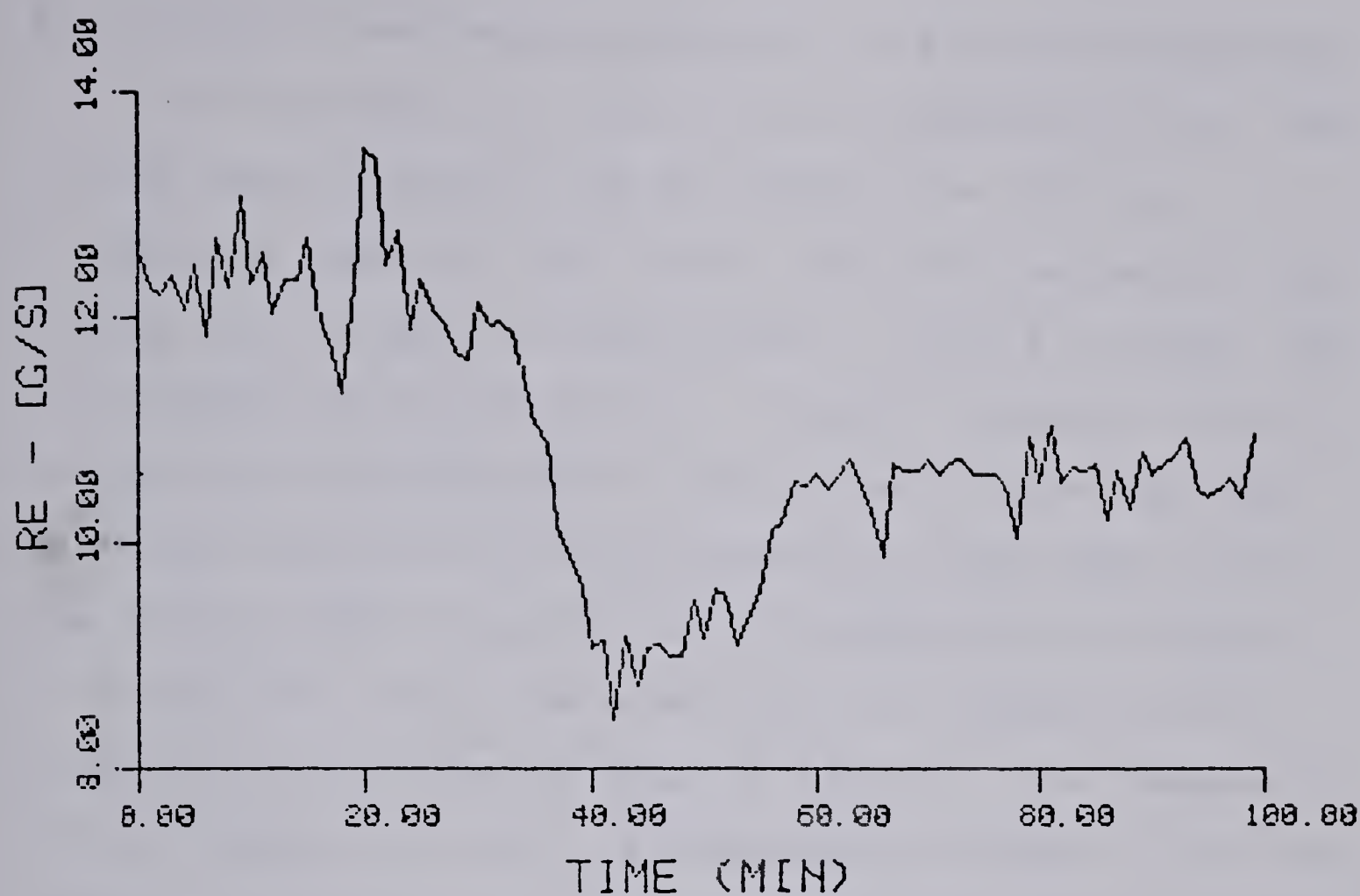


Figure 5.17 ST Control of Top Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT

5.5.2 Control Performance Using the STC Without Q-weighting

To illustrate the effect of the Q-weighting (Q-WT), the STC was used to control the top composition with the Q-weighting removed. The initial controller parameters and options are the same as those given in Table 5.2 except that the Q-weighting is set to zero. Figure 5.18 shows the top composition response when the controller is first applied while the variation of the parameters as they adapt during the initial identification period is displayed in Figures 5.19 and 5.20. As in the case with the Q-weighting, the parameters have converged after 30 samples. The response of the top composition for a -25% decrease in feed flow and the subsequent change to the normal steady state flow rate are shown in Figures 5.21 and 5.22. The SAE for these results and those of some duplicate runs are compared with the previous results obtained using Q-weighting in Table 5.4.

Table 5.4 Comparison of Top Composition Absolute Error Values (Over 60 Samples) for STC With and Without Q-weighting

Type of Disturbance	With Q-weighting		Without Q-weighting	
	Figure	SAE (wt%)	Figure	SAE (wt%)
-25% to Feed	5.16	2.177	5.21	4.278
Return to S.S.	5.17	2.197	5.22	2.693
-25% to Feed	Duplicate	2.347	Duplicate	2.906
Return to S.S.	Duplicate	1.867	Duplicate	3.327

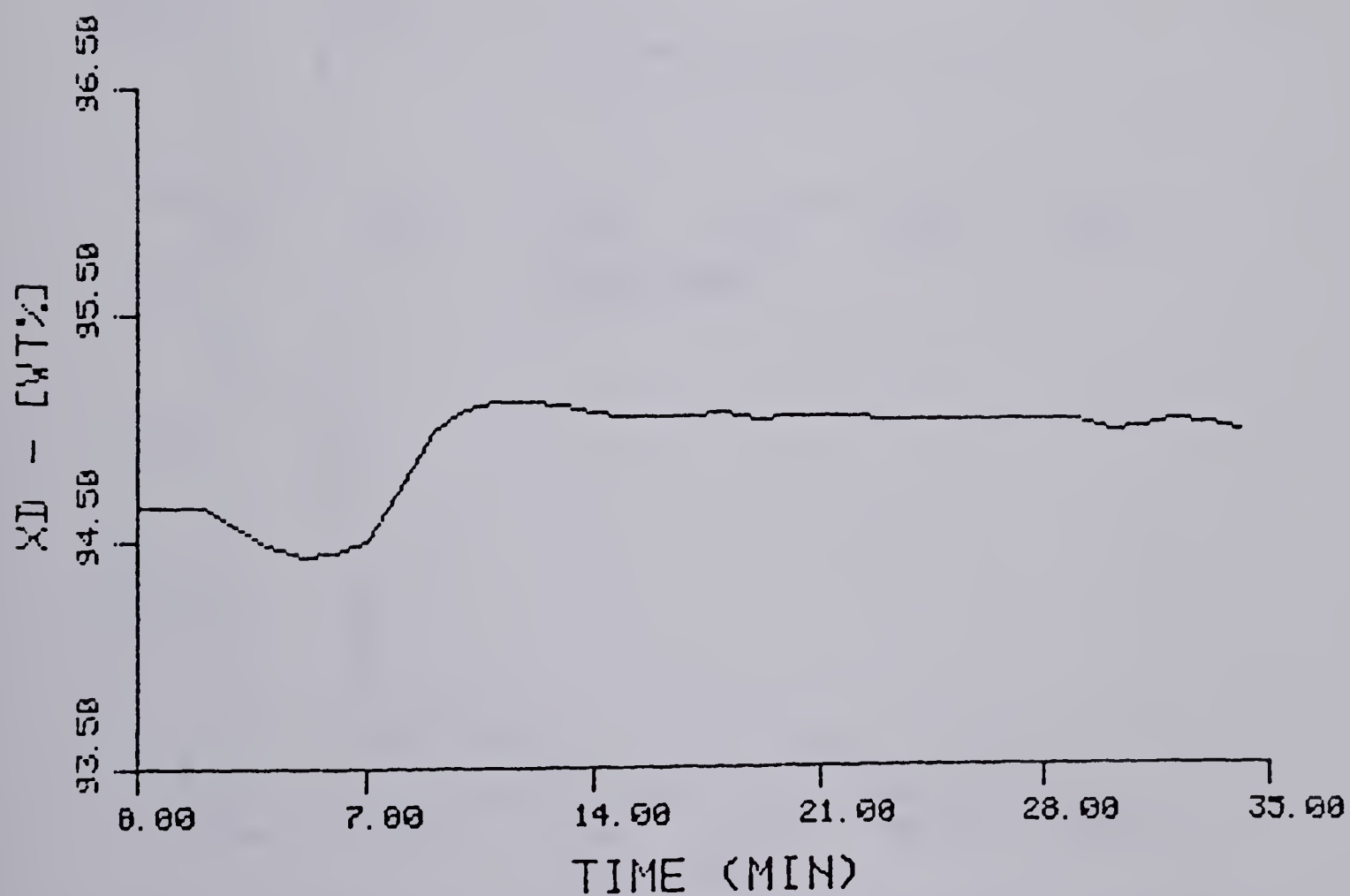
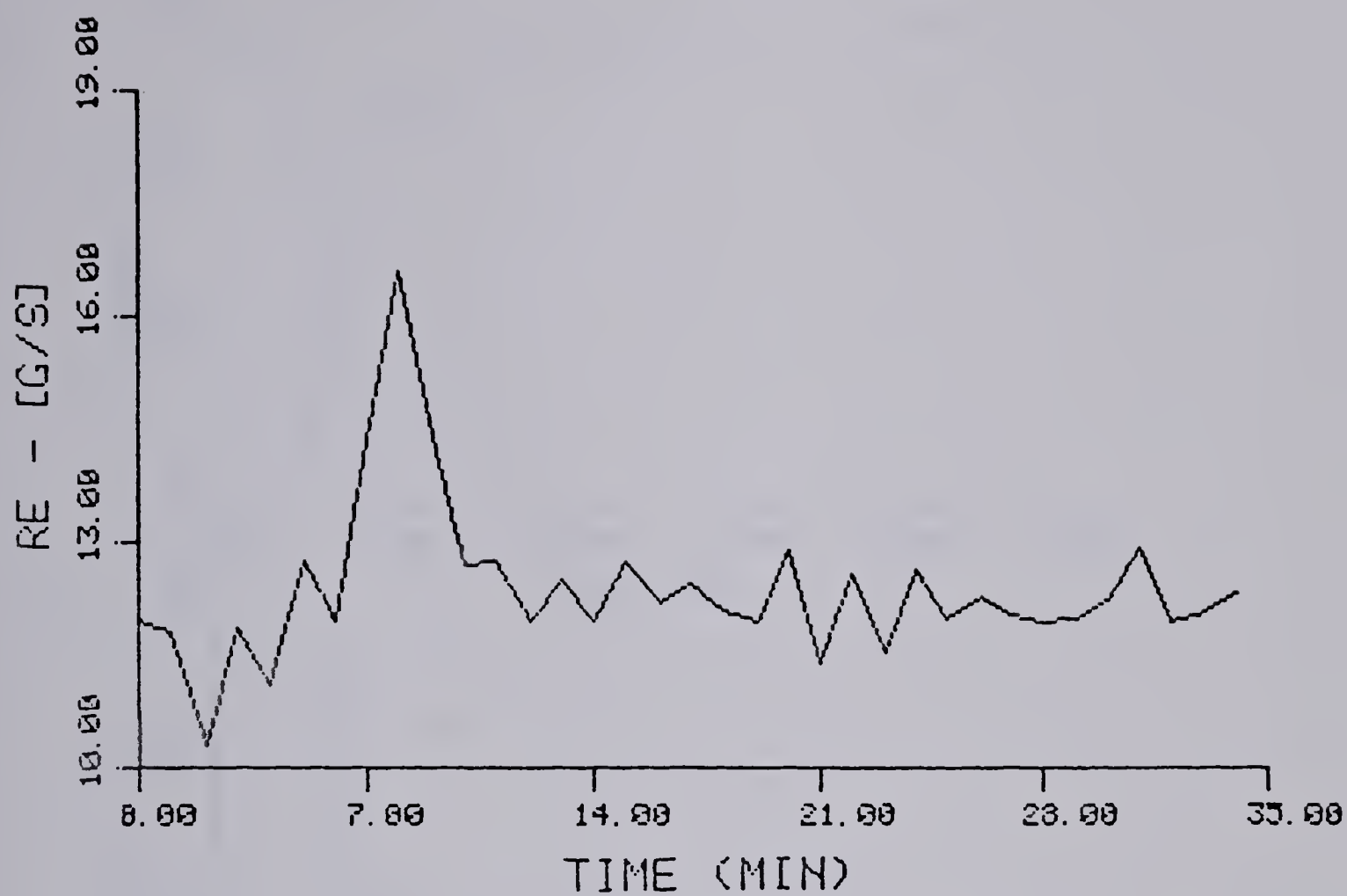


Figure 5.18 Top Composition Behaviour When Operation of the STC Without Q-WT is Initiated

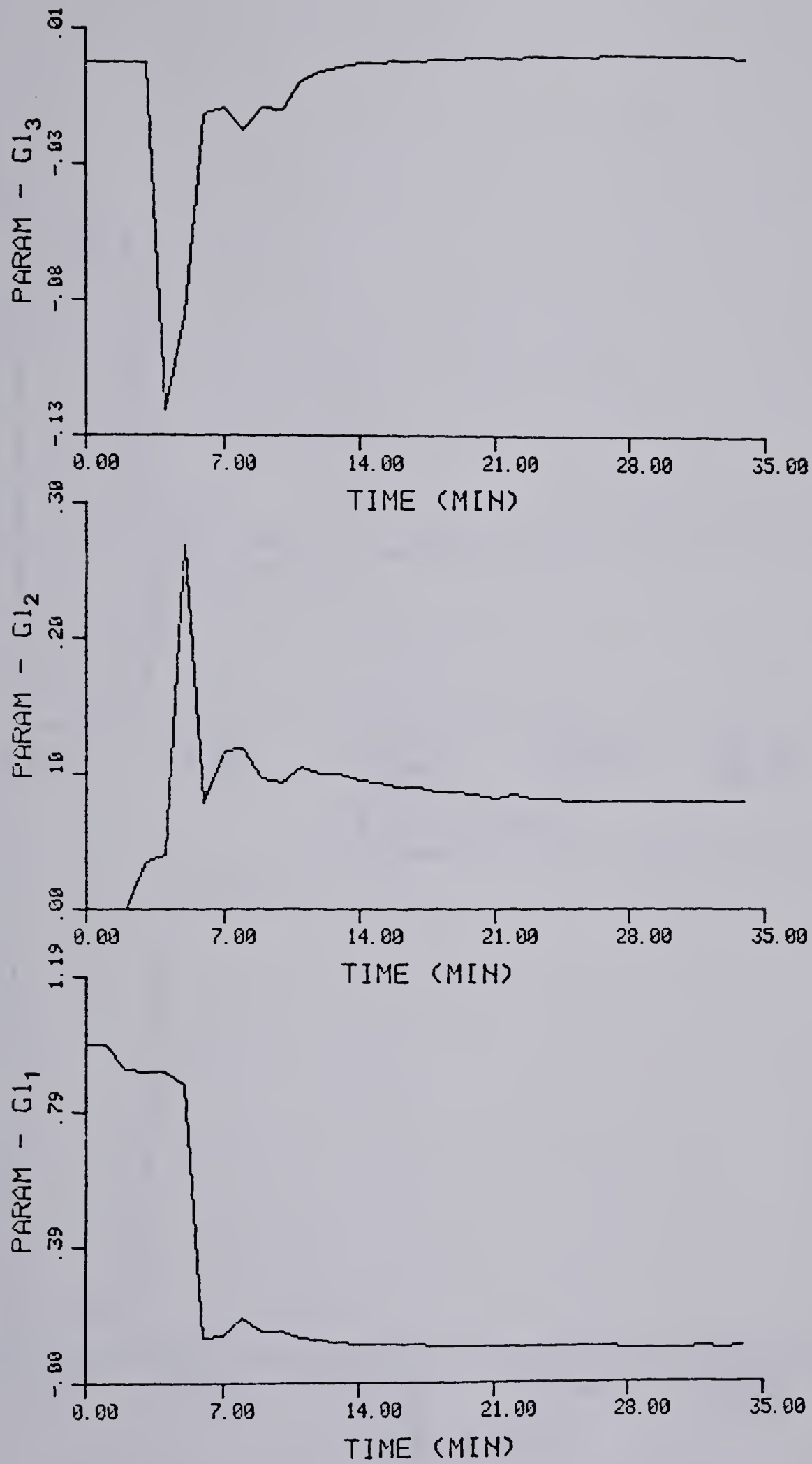


Figure 5.19 Variation of G1 Parameters in the Initial Tuning Period for STC Without Q-WT

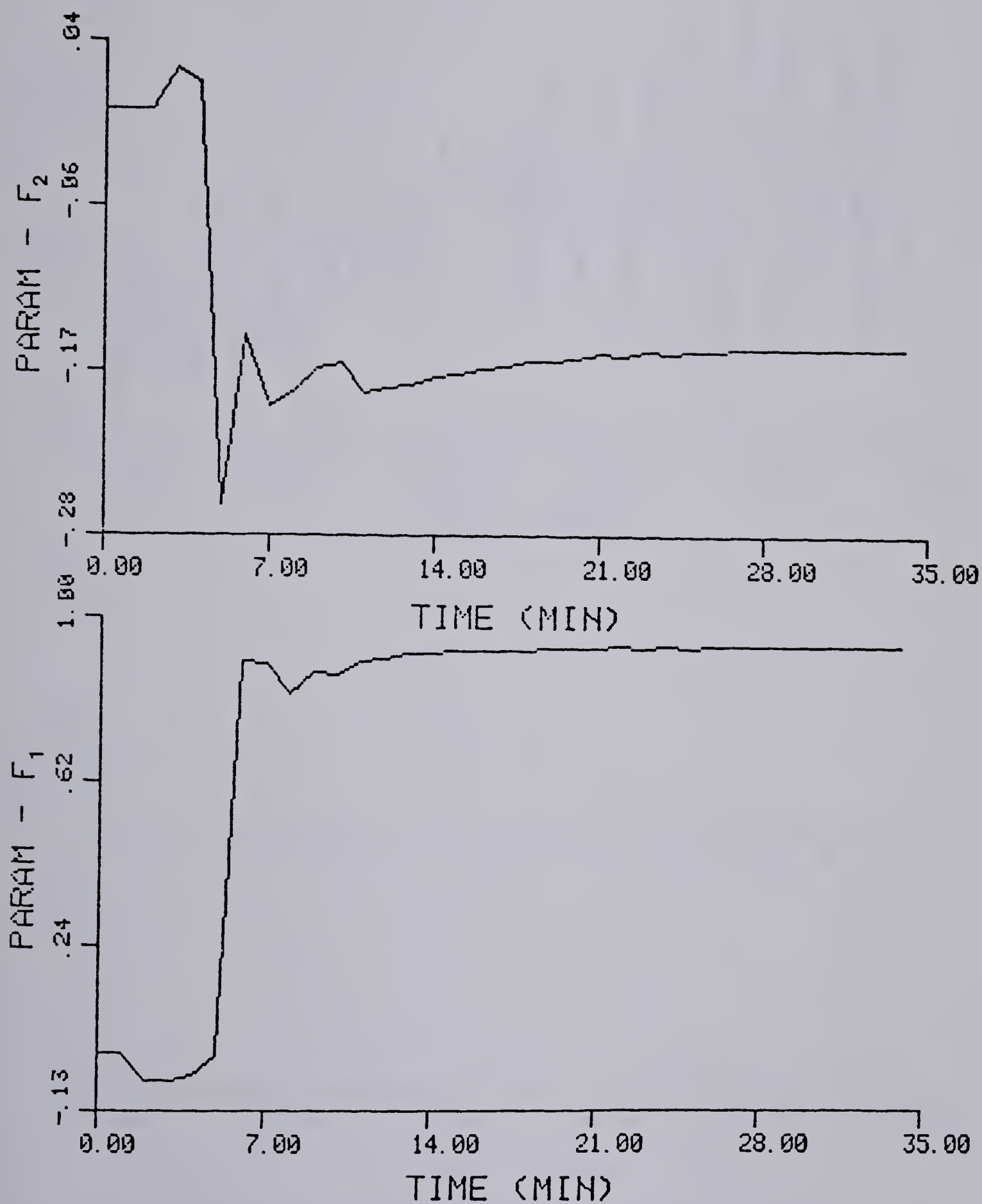


Figure 5.20 Variation of F Parameters in the Initial Tuning Period for STC Without Q-WT

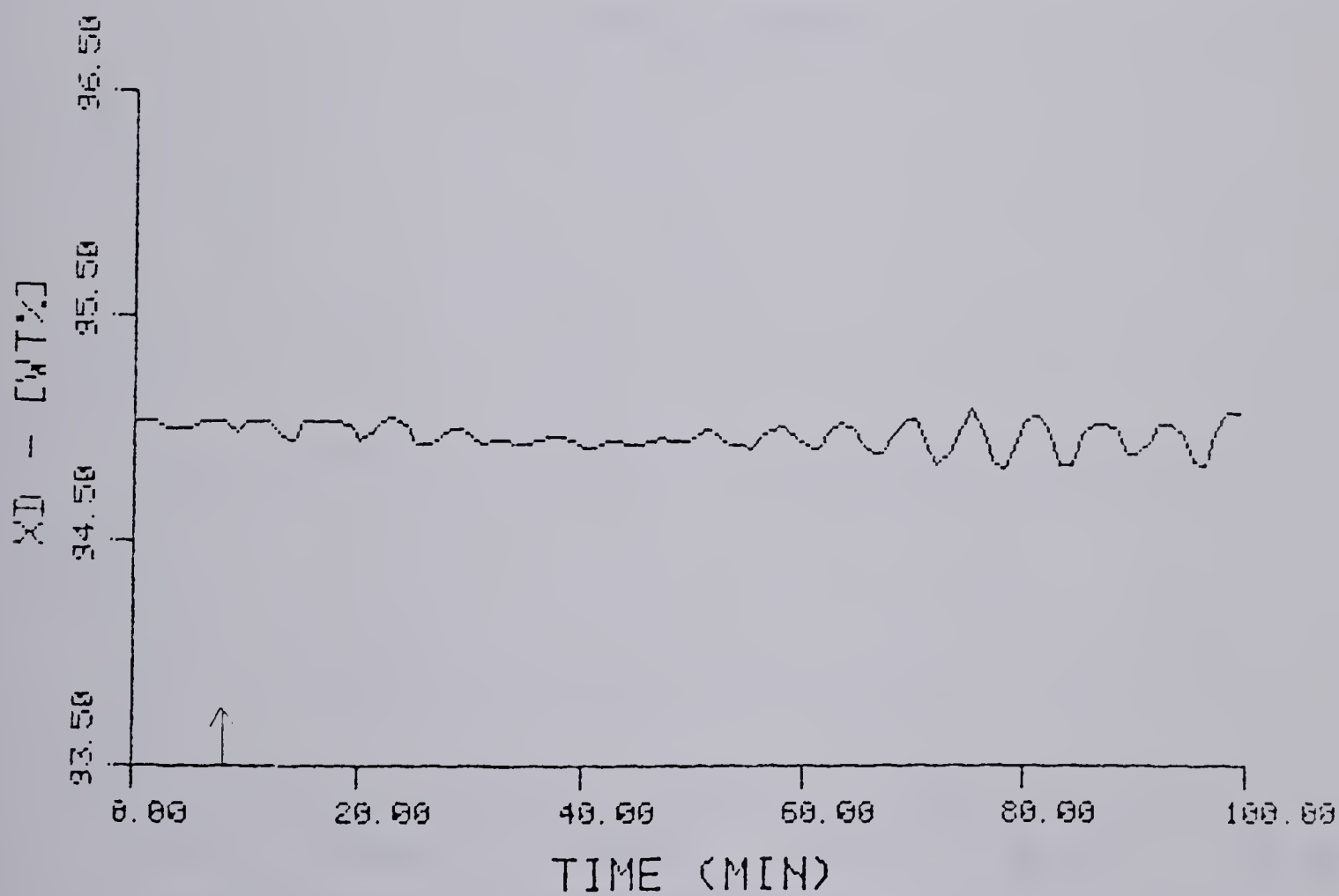
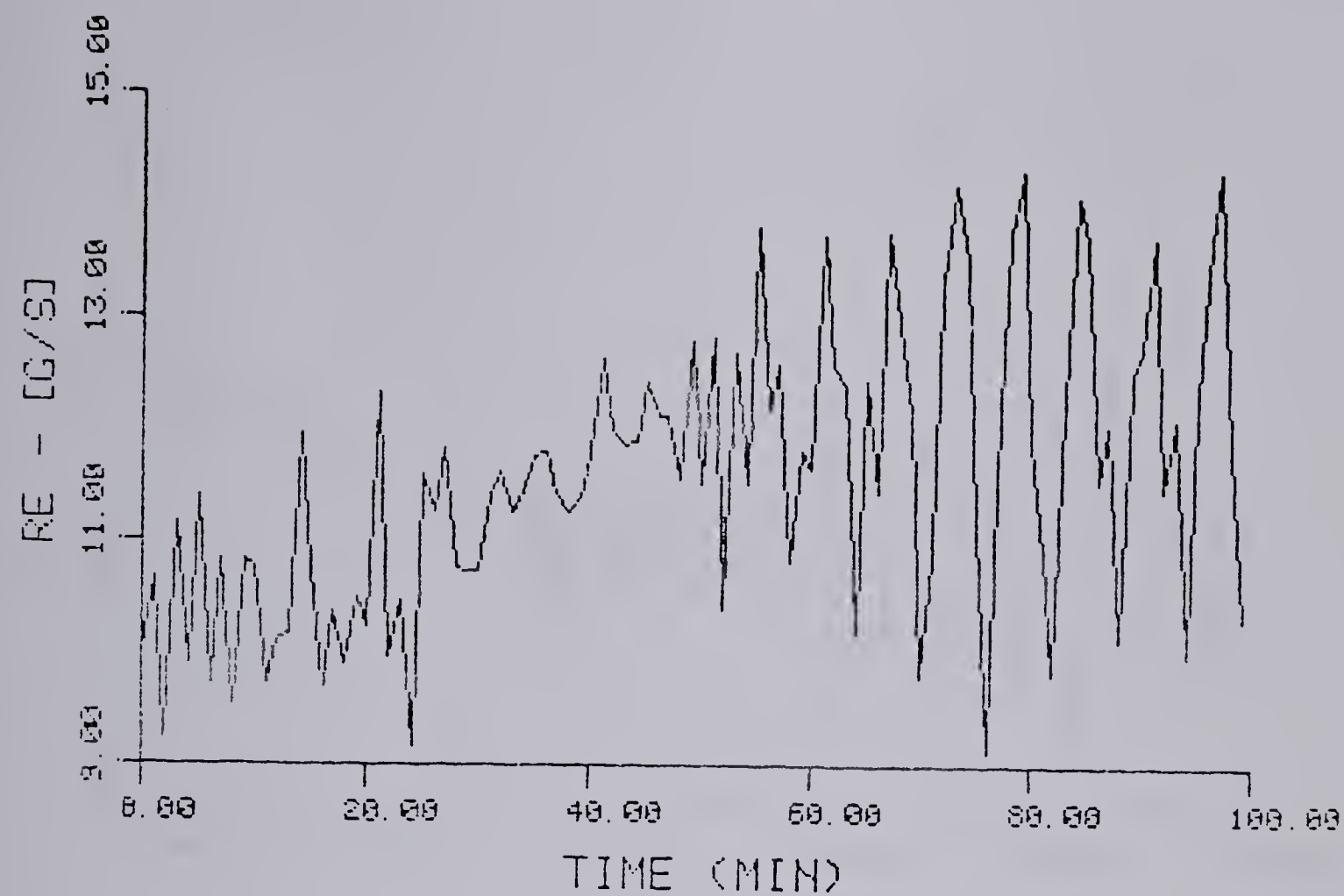


Figure 5.21 ST Control of Top Composition for a -25% Step Decrease in Feed Flow Rate Without Q-WT

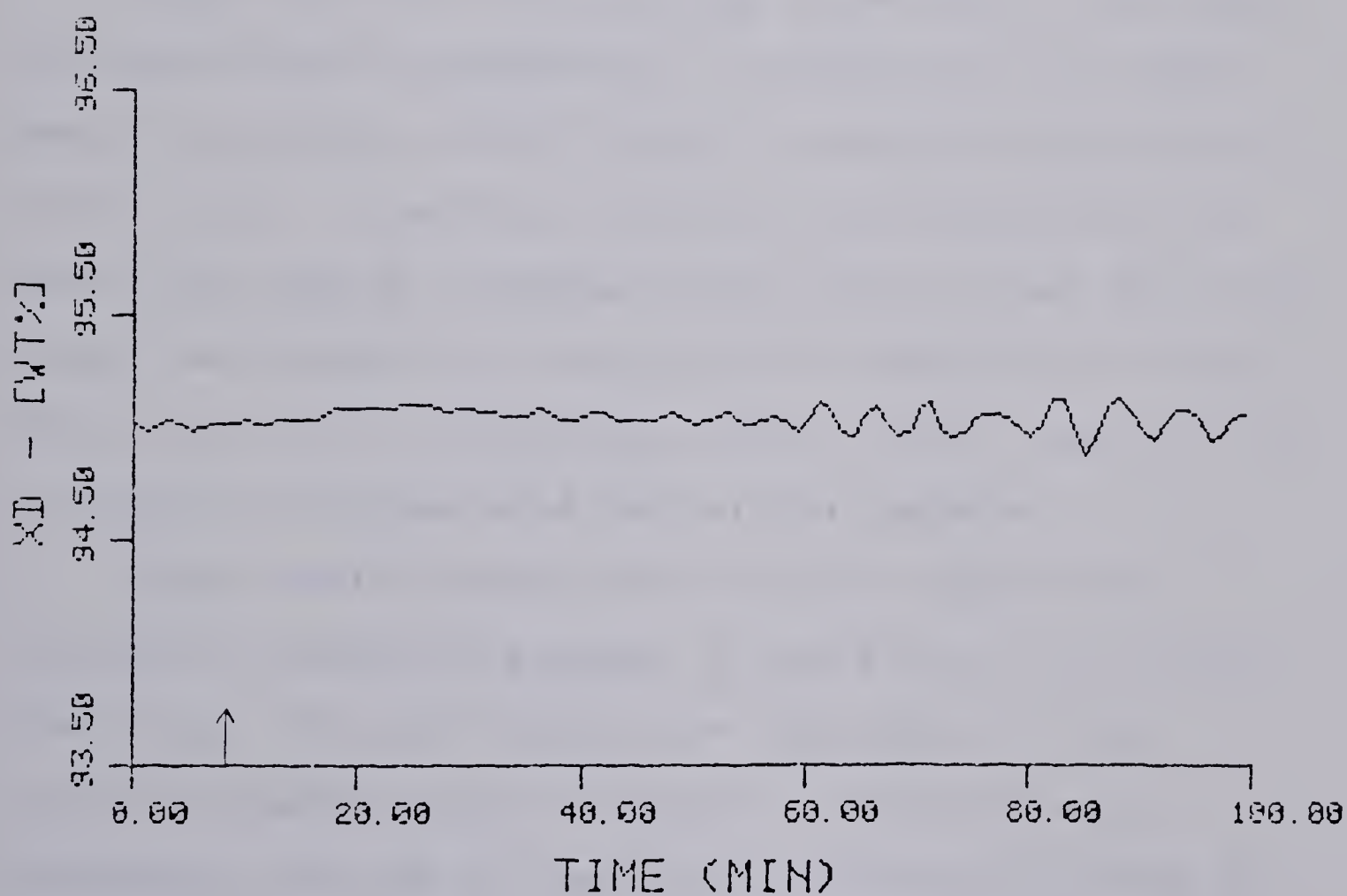
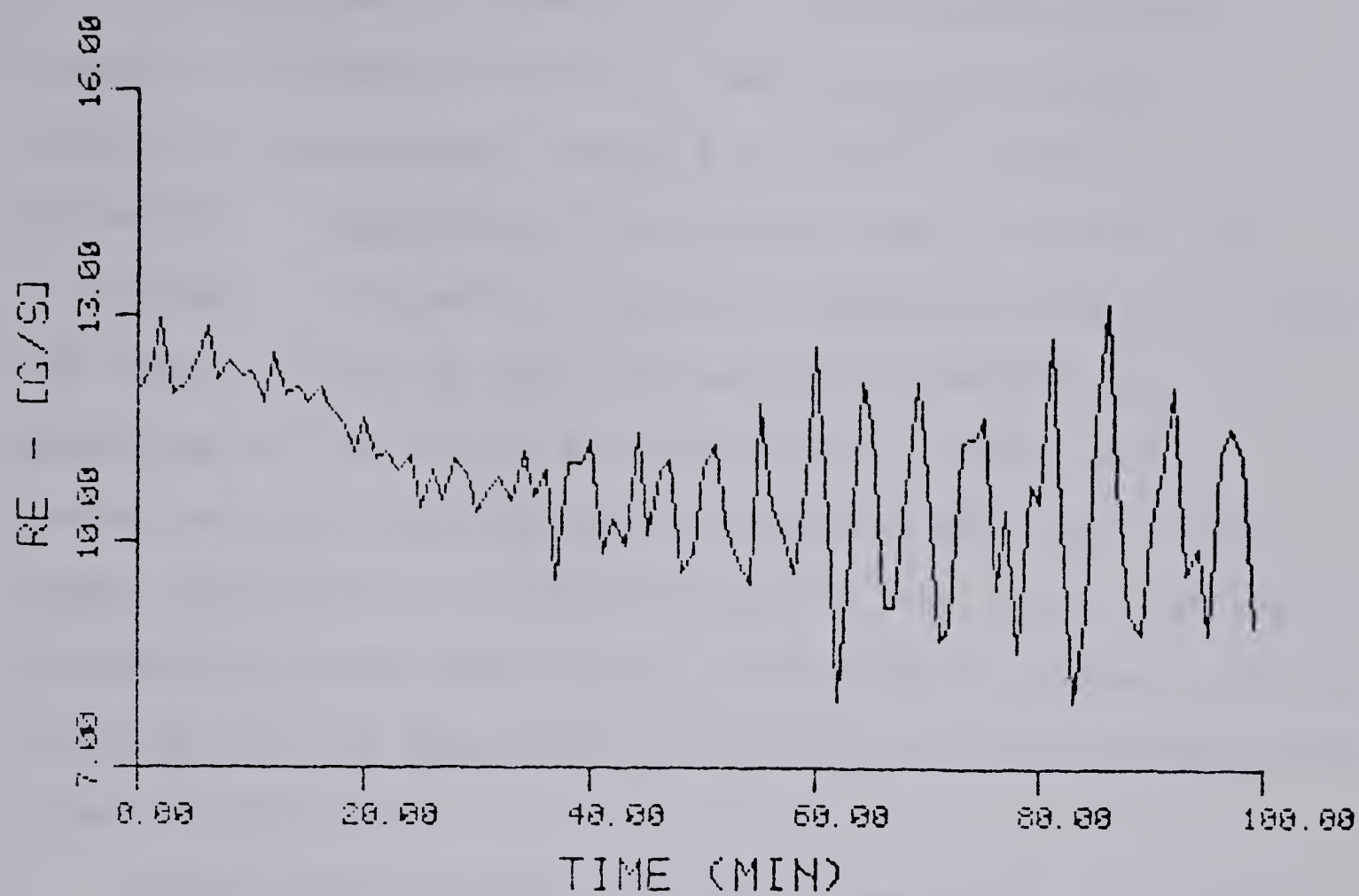


Figure 5.22 ST Control of Top Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value Without Q-WT

As discussed in Chapter Two, the Q-weighting adds a penalty for control effort so removing the Q-weighting allows the manipulated variable to exhibit larger deviations. Comparison of the variations in reflux flow rate shown in Figure 5.21 and 5.16 shows that this is indeed the case. It can be seen that without Q-weighting, the magnitude of the reflux flow variation is twice the variation that resulted when Q-weighting was used. The larger variation in reflux flow due to the removal of the Q-weighting causes the control performance to deteriorate as shown by the SAE values for STC with and without Q-weighting given in Table 5.4.

Comparing the initial tuning in period for the cases with and without Q-weighting, it is seen that the added penalty in control effort helps to reduce the variation in reflux flow. In addition, without the Q-weighting, the controller demands a maximum reflux flow of over 16 g/s but using the Q-weighting, the maximum is reduced to 11.5 g/s. This reduction in the maximum control output leads to less overshoot in the overhead composition response.

These results clearly show that the use of the Q-weighting reduces the amount of variation in the reflux flow rate. This will lead to an improvement in the controller performance in terms of the SAE criteria. Furthermore, the use of Q-weighting does not slow down the rate at which the parameters converge during the initial tuning in period and it gives the added advantage of

reducing the maximum reflux flow required during the initial transition period.

5.5.3 Effect of Sampling Time

To show the effect of sampling time, experimental runs were made using times of two and three minutes. The experimental responses for a -25% step decrease in feed flow rate, for the two and three minute rates are shown in Figures 5.23 and 5.24. With the exception of the change in sampling time and the absence of the Q-weighting, the initial controller parameters and options are the same as those shown in Table 5.2. No Q-weighting was employed in order to demonstrate that increasing the sampling time has the same effect as adding the Q-weighting; that is in terms of reducing the oscillation in the manipulated variable[24].

Comparing the overhead composition response for the different sampling times, it is seen that by increasing the time to two minutes, the magnitude of the variation in the reflux flow is greatly reduced. As can be seen from the absolute error values in Table 5.5, the performance is not much better than that achieved using the one minute sampling rate because at the two minute sampling time, the controller responds more slowly to an unknown disturbance since the controller is only acting in the feedback mode. By further increasing the sampling time to three minutes, the magnitude of the oscillation in the reflux flow starts to increase. The controlled response to the disturbance is not as

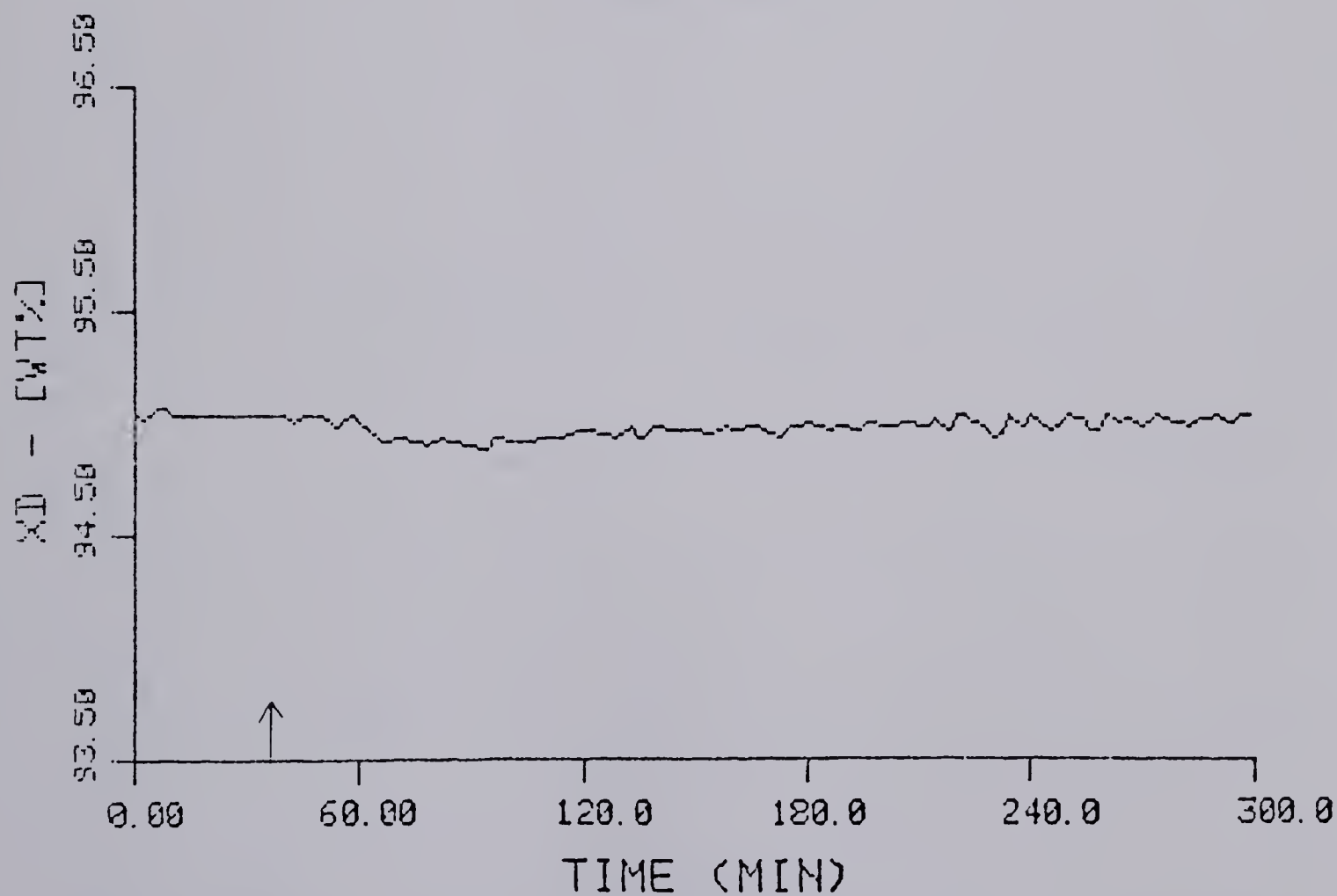
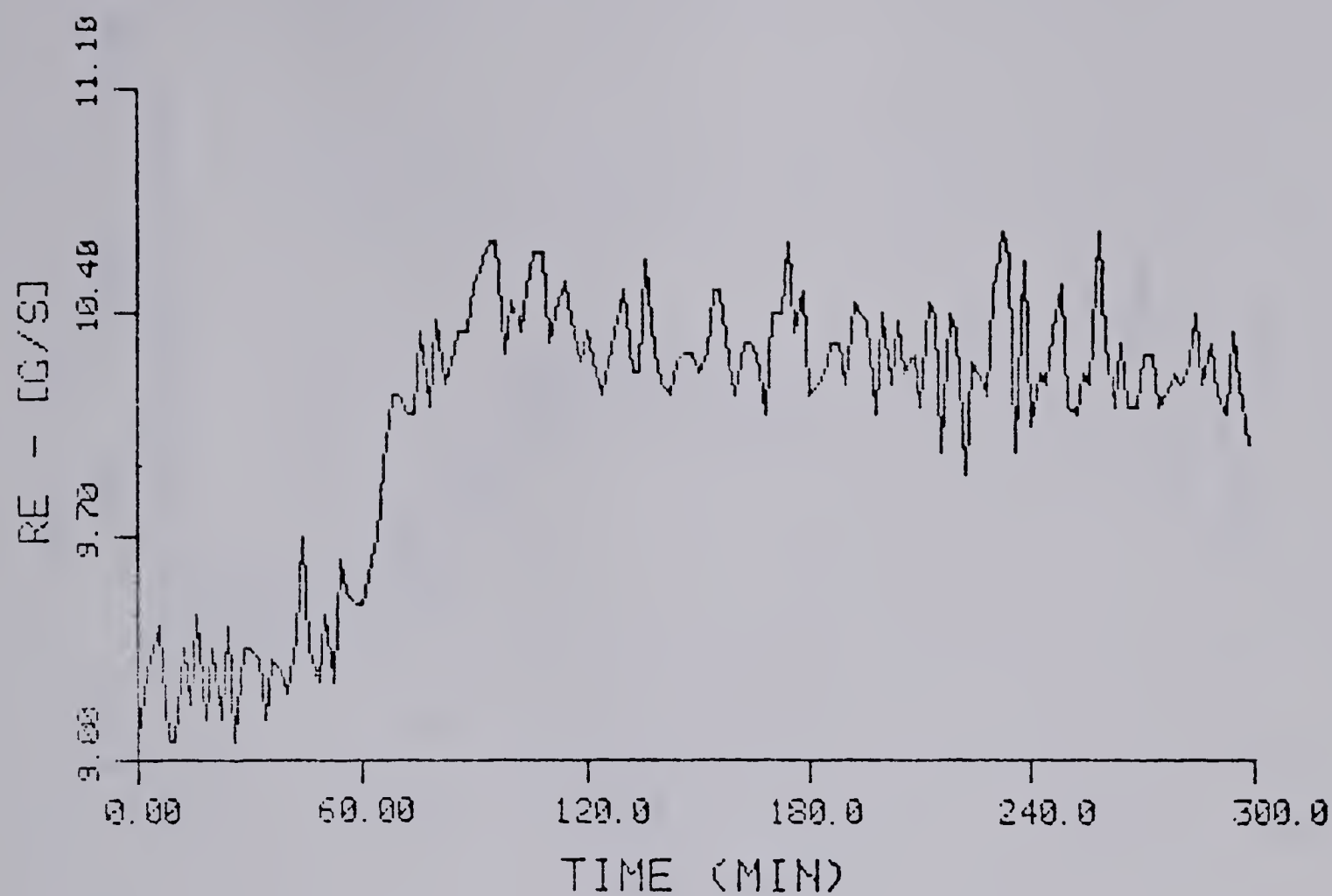


Figure 5.23 ST Control of Top Composition to a -25% Step Decrease in Feed Flow Rate Without Q-Wt: Two Minute Sampling Time

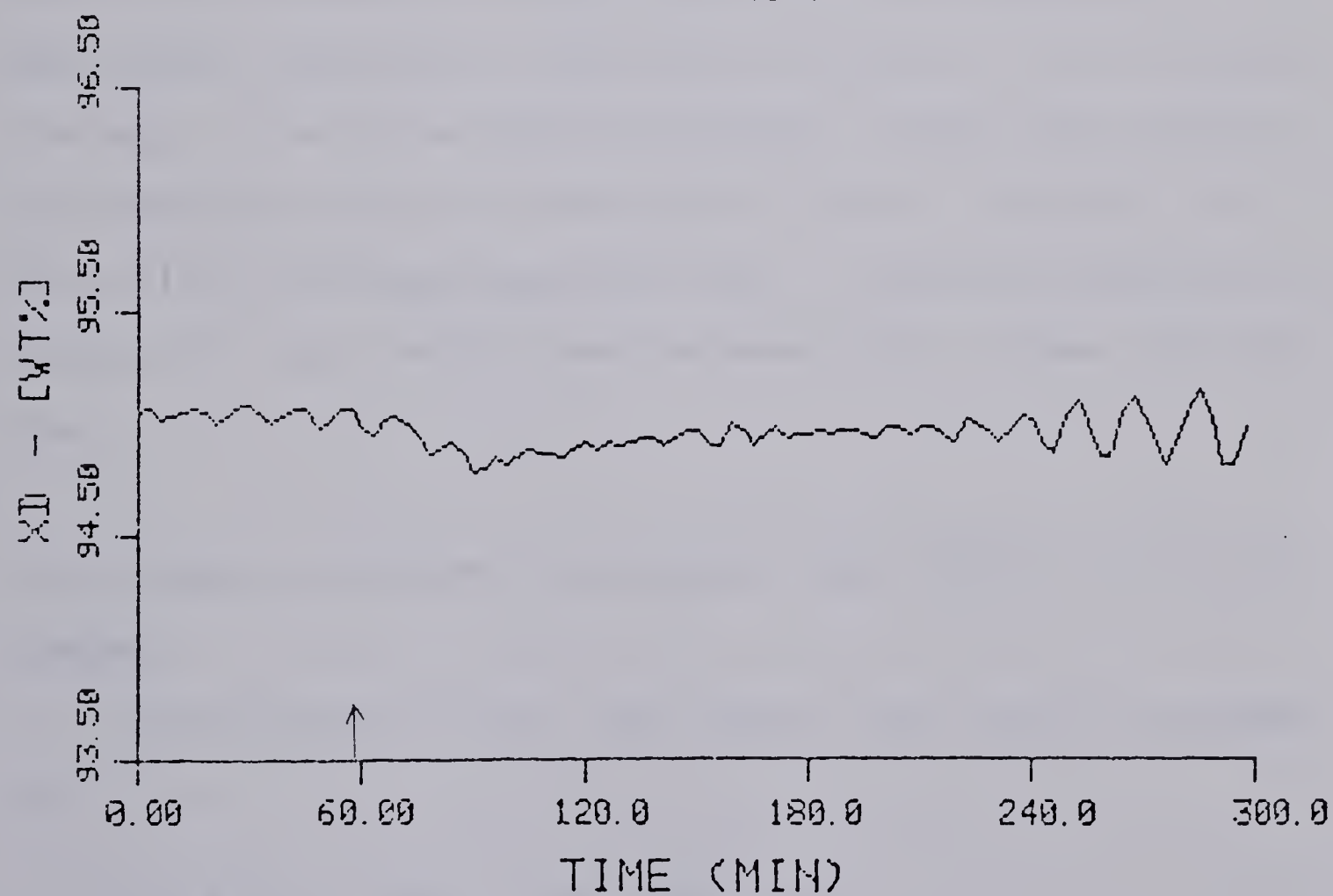
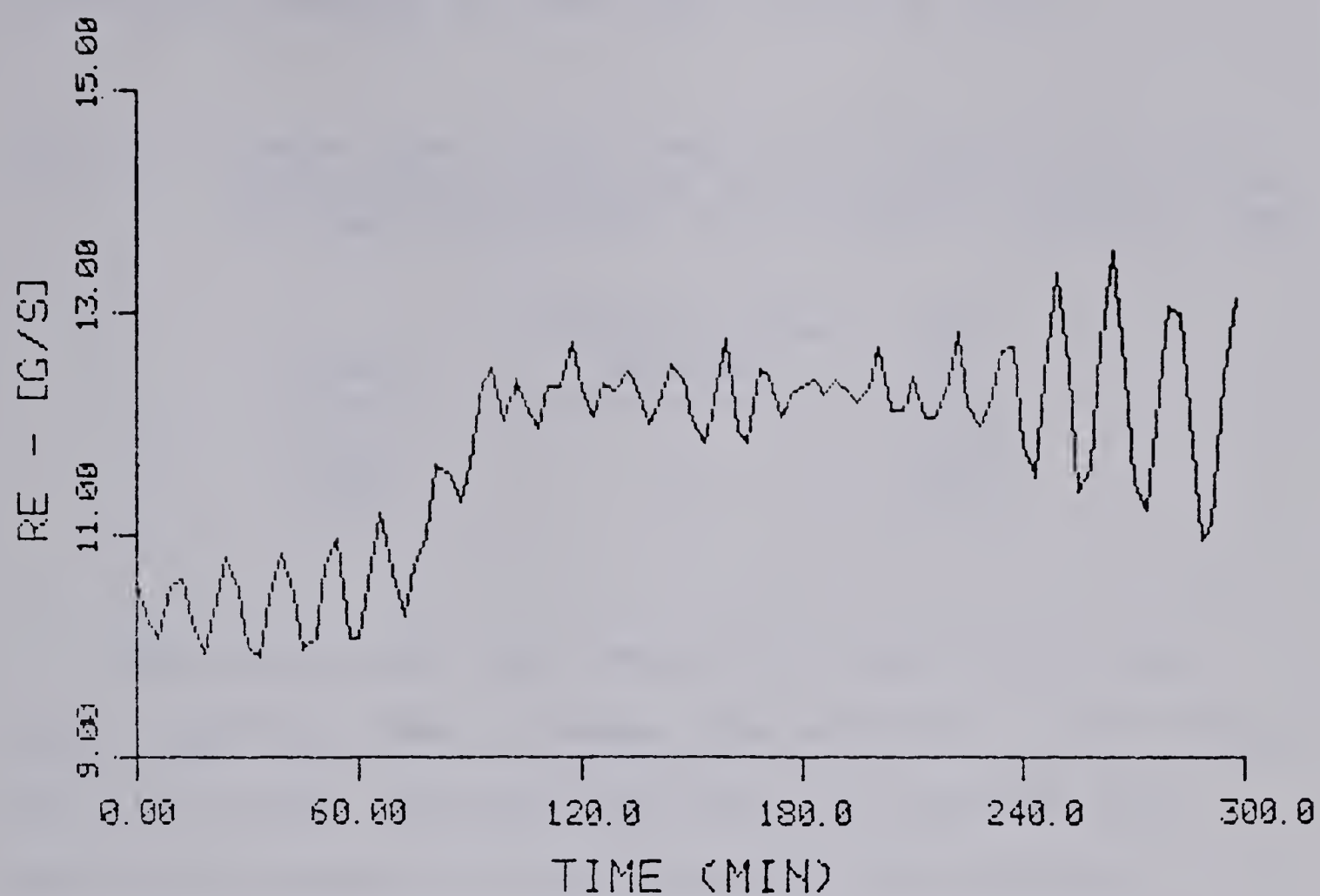


Figure 5.24 ST Control of Top Composition to a -25% Step Decrease in Feed Flow Rate Without Q-Wt: Three Minute Sampling Time

satisfactory because of the larger sampling time.

Table 5.5 Comparison of Top Composition Absolute Error Values Calculated for a Period of One Hour After the Feed Disturbance for Different Sampling Times

Figure	Sampling Time (minutes)	SAE (wt%)
5.16	1	4.278
5.23	2	3.627
5.24	3	6.700

The results show that while it is possible to use larger sampling times to reduce the variation in the reflux flow, the larger sampling time tends to slow down the controller response for a disturbance that has to be detected by feedback of the output variable. For a feedback strategy, it would be better to sample faster and increase the weighting placed on the control effort. However, for a controller with feedforward action, it should be possible to achieve the same control performance with a larger sampling time.

5.5.4 Effect on Control Performance when Identification is Stopped

From Equation (2.34), the control law can be expressed as:

$$F\bar{y}_{t+k} + Gu_t - HQu_t + H\bar{w}_t + D\bar{v}_t = 0 \quad (5.1)$$

and without feedforward action, the control law reduces to:

$$F\bar{y}_{t+k} + Gu_t - HQu_t + H\bar{w}_t = 0 \quad (5.2)$$

At steady state, the outputs and inputs remain constant and Qu_t tends to zero (cf. Chapter 2). Equation (5.2) becomes:

$$\sum F_i \bar{y}_{t+k} + \sum G_i u_t - \sum H_i \bar{w}_t = 0 \quad (5.3)$$

If the identification scheme is stopped, the sum of the coefficients for each of the polynomials becomes constant. When a disturbance is introduced, it is clear that the control output will reach a new steady state value. Since the sum of these coefficients remains the same, to satisfy Equation (5.3), only the process output value can change hence resulting in an offset. The magnitude of the offset will depend on the G and F parameters. If the sum of the G parameters is small compared to the sum of the F parameters, only a small change in \bar{y}_{t+k} is needed, so a smaller offset occurs.

To demonstrate the effect discussed, a run was made where the identification was stopped after the STC had brought the top composition to its steady state value. The initial controller parameters and options are the same as those given in Table 5.2 with the exception that Q-weighting is not used. The response of the top composition to a -25% step change from the steady state feed flow rate is recorded in Figure 5.25. Although the offset is small, it is clear that there is an offset of about 0.05%. By resuming the

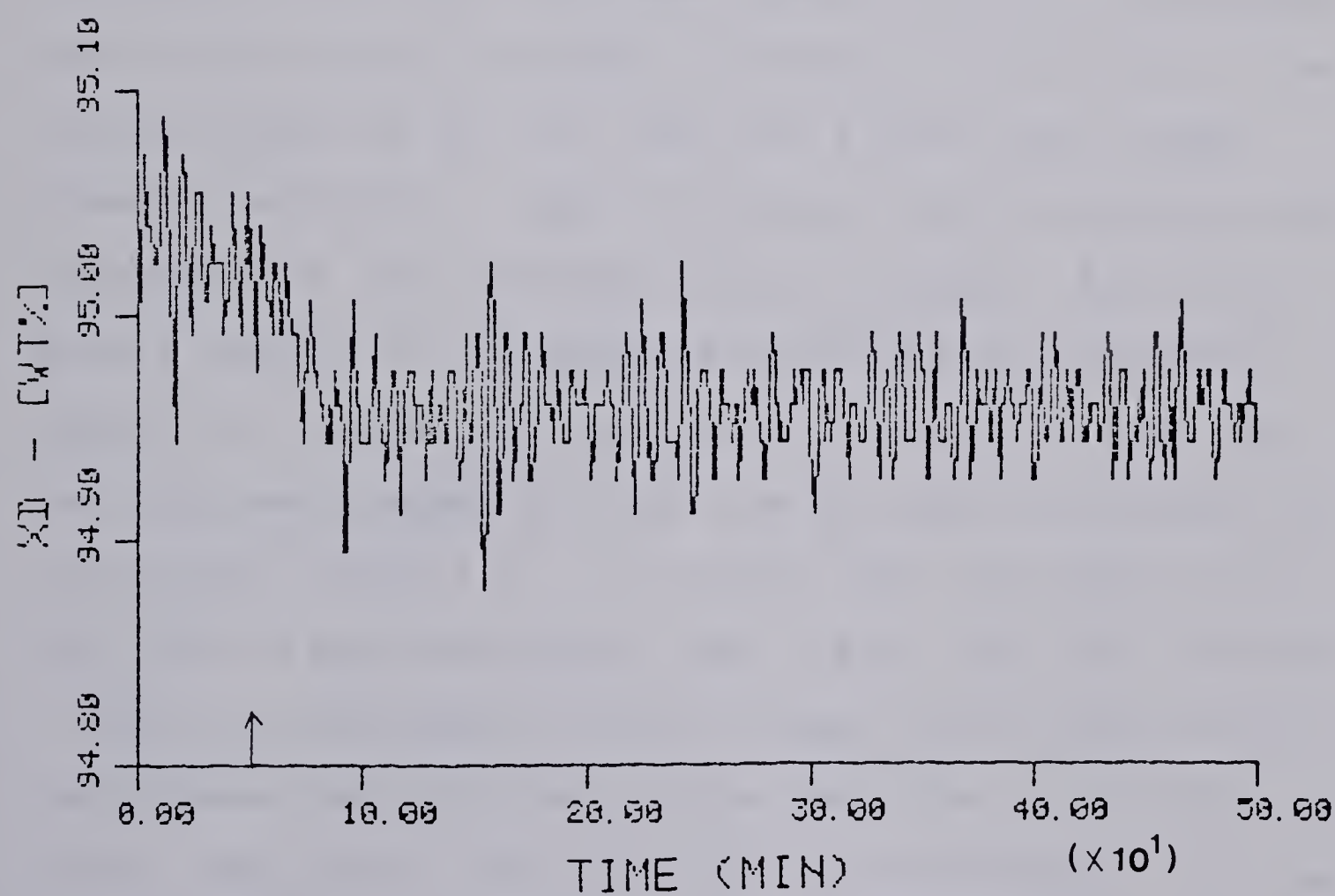
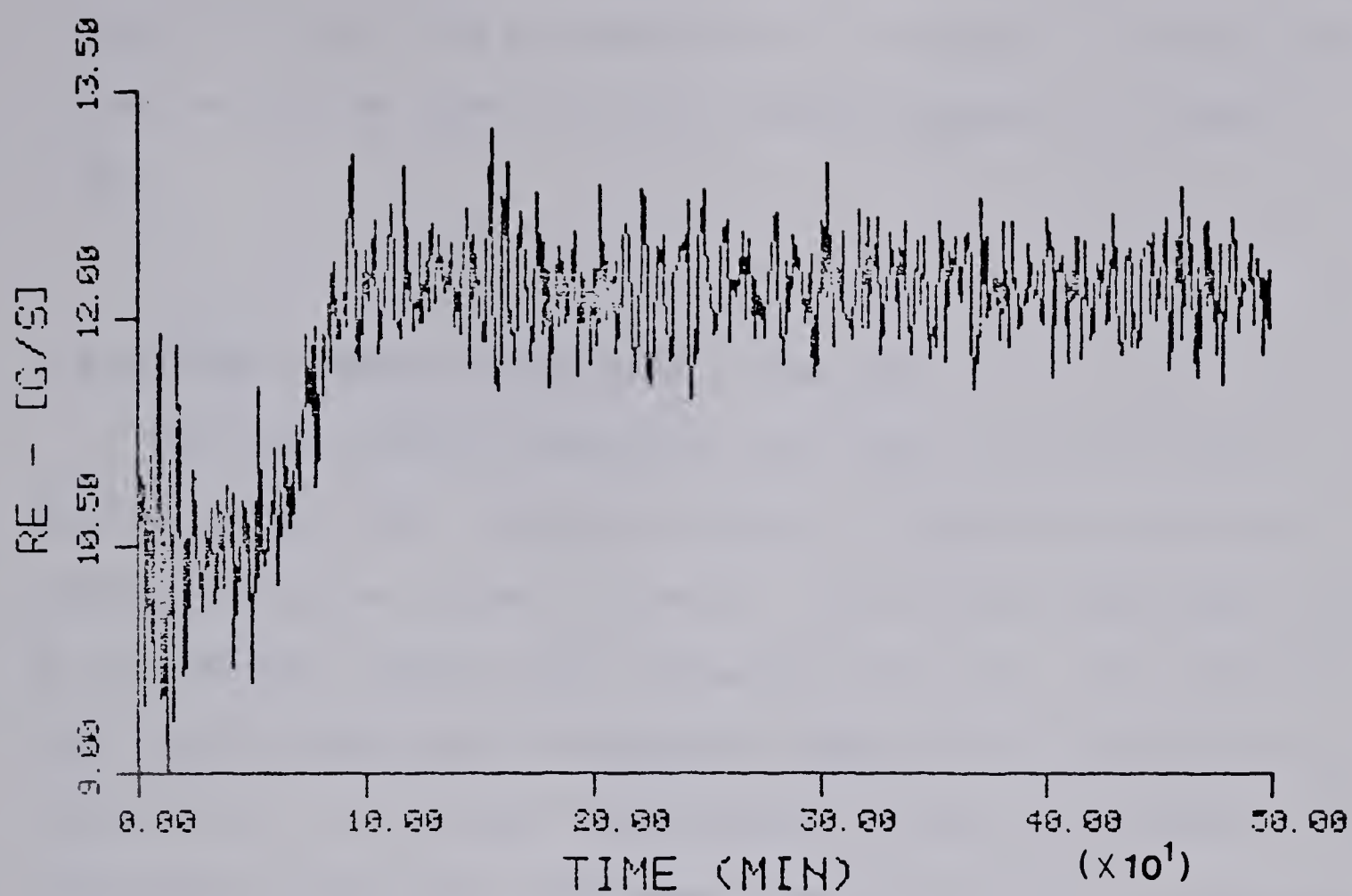


Figure 5.25 ST Control of Top Composition to a -25% Step Change in Feed Flow Rate Without Q-WT and the Identification Stopped

identification, the parameters will readjust to remove the offset as can be seen from the response shown in Figure 5.26.

5.6 BOTTOM COMPOSITION CONTROL WITH STC

The self-tuning controller was used to control the bottom composition, despite changes in feed flow rate, by manipulating the steam flow rate. The reflux flow was fixed at its normal steady state value of 10.2 g/s. The initial self-tuning controller parameters and options are given in Table 5.6. The control behavior for bottom composition that resulted for the four different feed flow rate disturbances used in this study are shown in Figures 5.27 to 5.30. The results given are for STC runs that already have 'good' parameter estimates. Some re-tuning of the Q-weighting was necessary with the resulting values as noted. Table 5.7 gives a summary of the absolute error values for bottom composition control for a period of 1 1/2 hour after the feed rate was changed for both the STC and the previous PID results (cf. Table 5.1). It can be seen from these values that the STC performs better than PID for the +25% increase in feed flow rate away from the steady state value and for the decrease back to steady state feed flow, while PID is better than the STC for the other two disturbances. On the basis of these results, it can be seen that both controllers performed equally well. However, it should be noted that considerable more time was needed to tune the PID

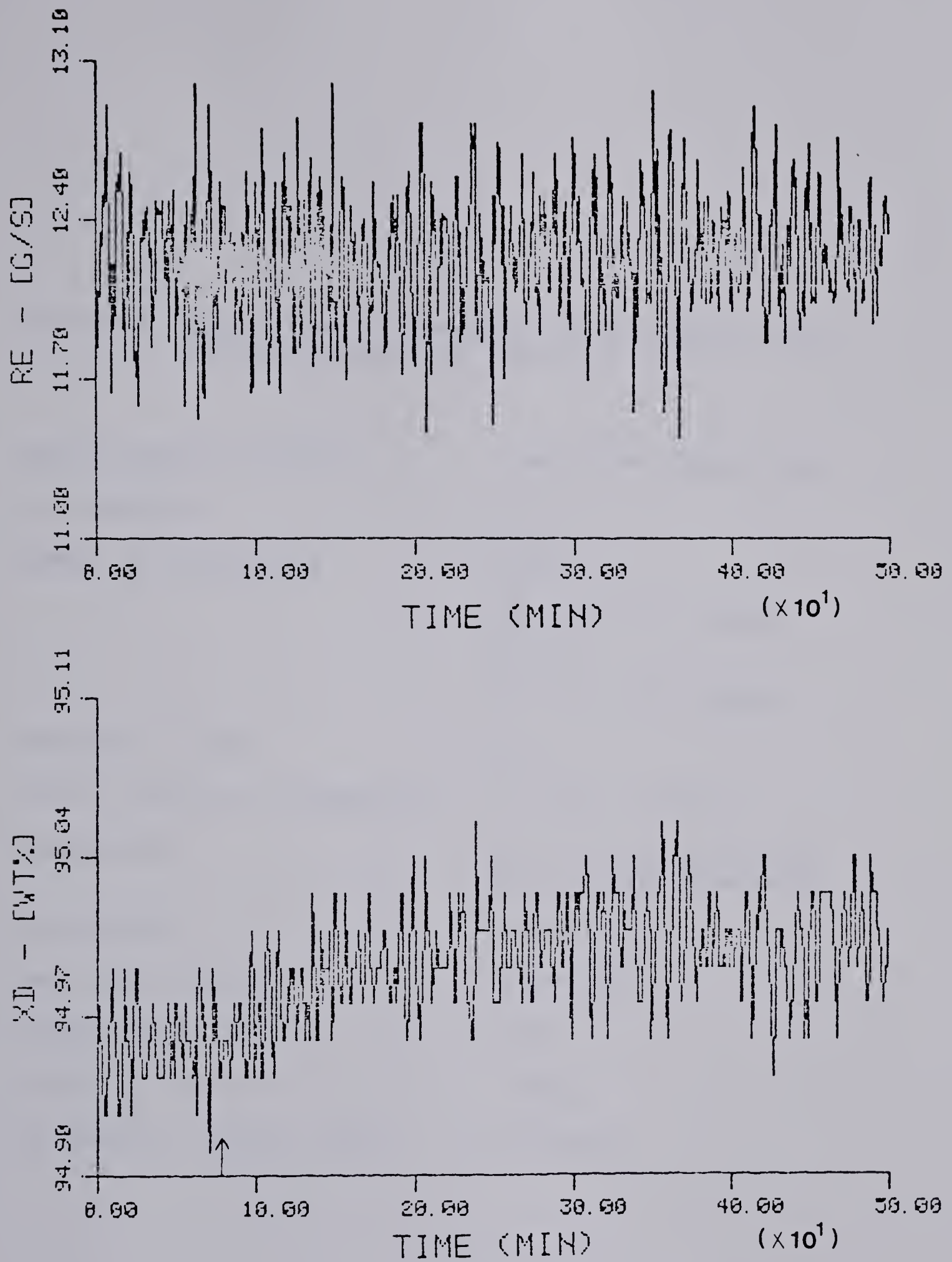


Figure 5.26 Top Composition Response After Resuming Identification in the STC

Table 5.6 Controller Parameters and Options for STC Bottom Composition Control

Identification Scheme	: Recursive Square Root
G Polynomial	: EB
Number of Parameters	: NG1 = 4 NF = 3 ND = 5 (FF added) NH = 1 K1 = 1 K3 = 1 (FF added)
Parameter Fixed	: $h_0 = -1$
Initial Value of Parameters	: All set to zero
Q-weighting	: $k_p = 1.145 \text{ (g/s)/wt\%}$ $k_i = 0.226 \text{ (g/s)/wt\%}$
P-weighting	: Set to 1
Sampling Interval	: 3 minute
Forgetting Factor	: 0.99
Initial \underline{p}^C Matrix	: $1000\underline{I}$
Incremental Control Limit	: 2.25 g/s

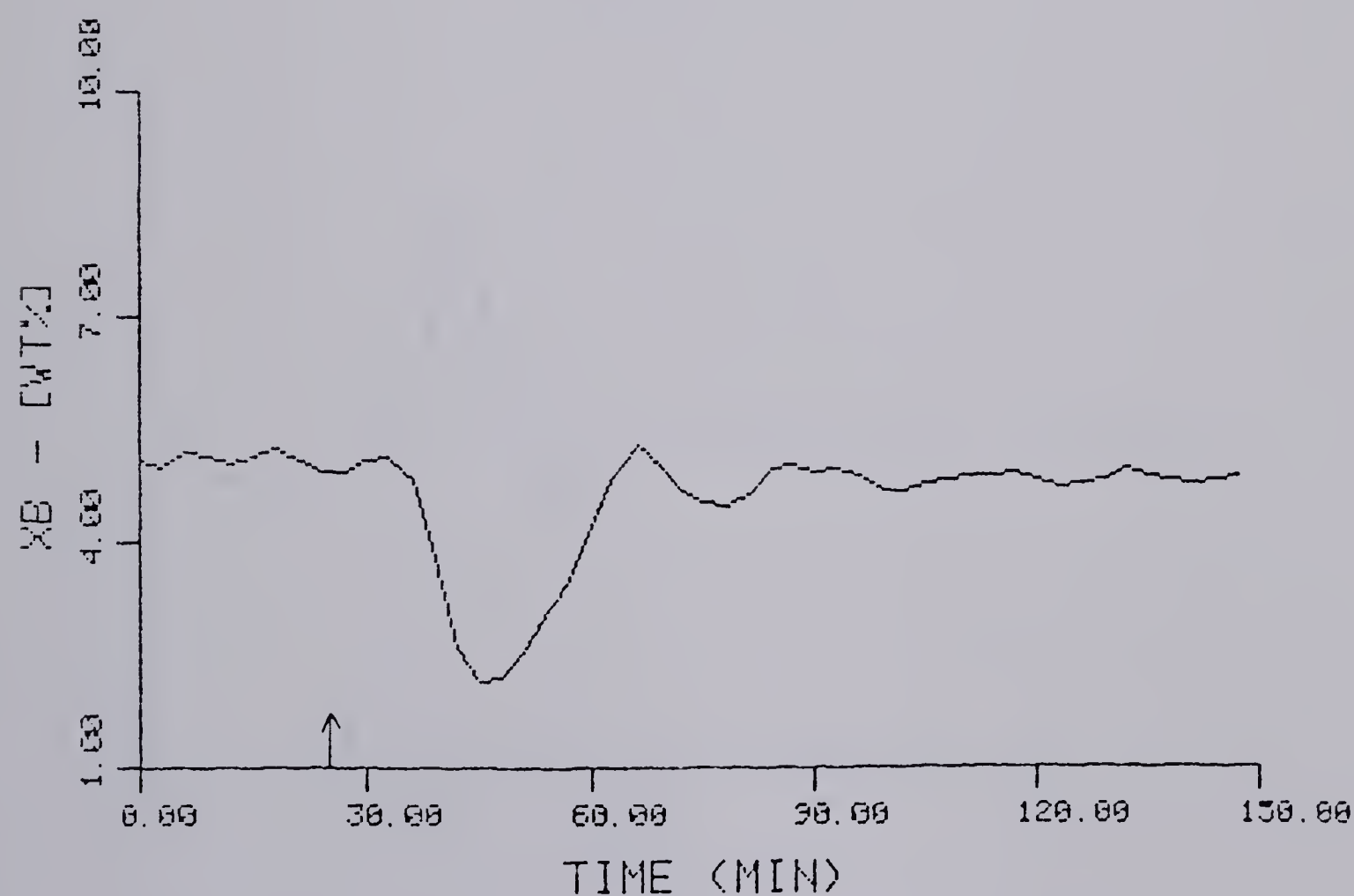
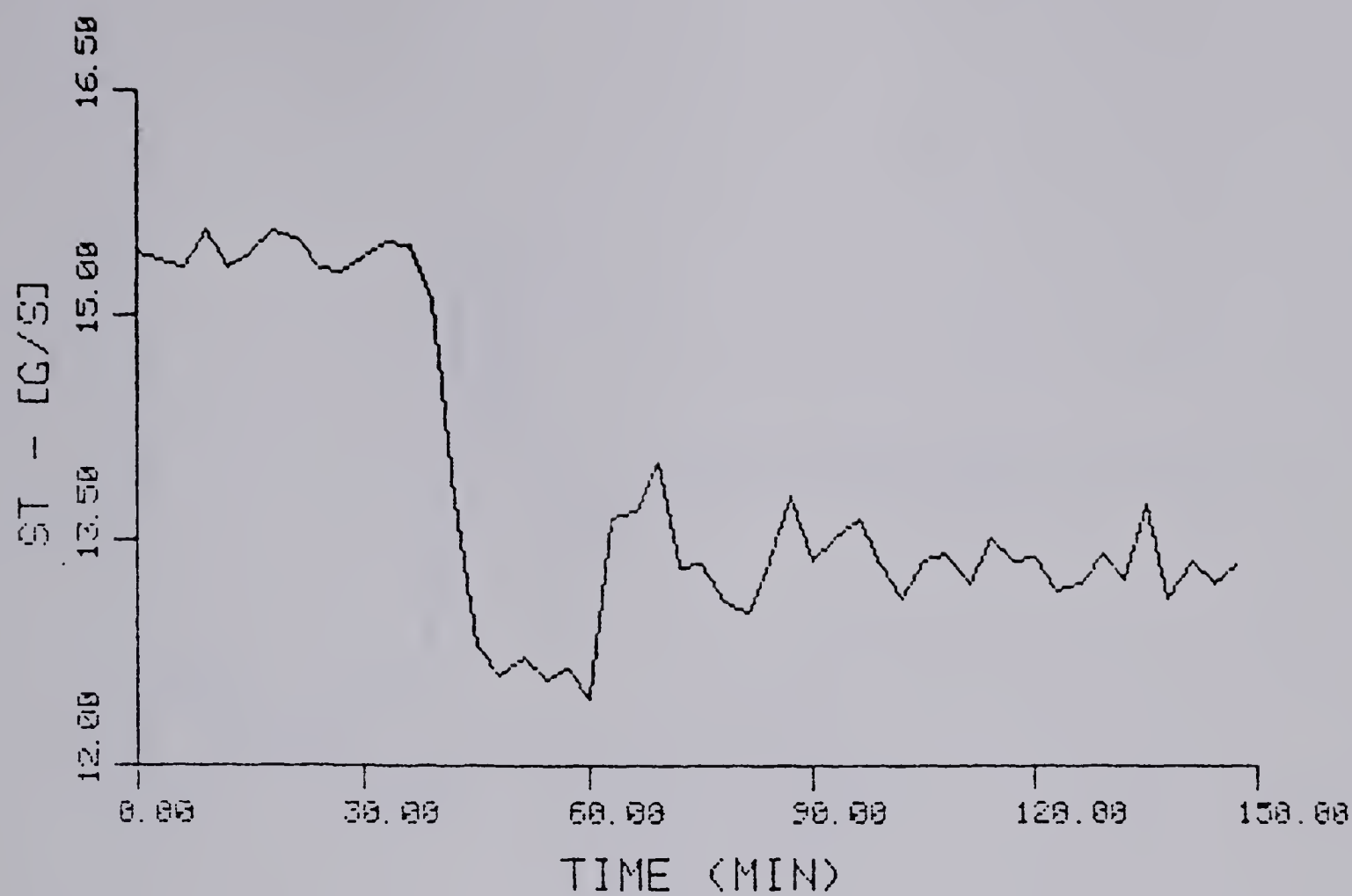


Figure 5.27 ST Control of Bottom Composition for a -25% Step Change in Feed Flow Rate With Q-WT ($k_p = 1.15$ and $k_i = 0.23$)

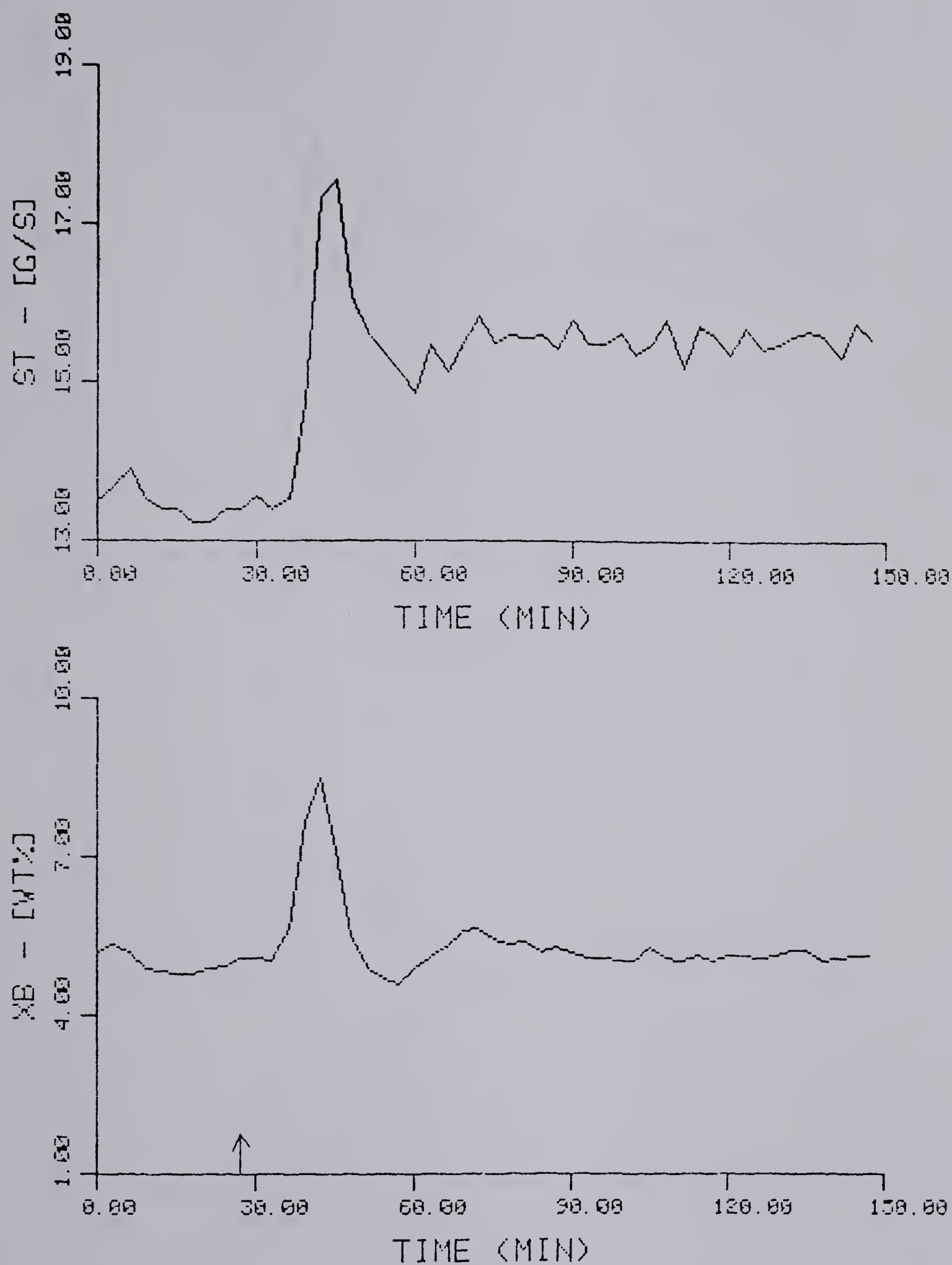


Figure 5.28 ST Control of Bottom Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT ($k_p = 0.90$ and $k_i = 0.47$)

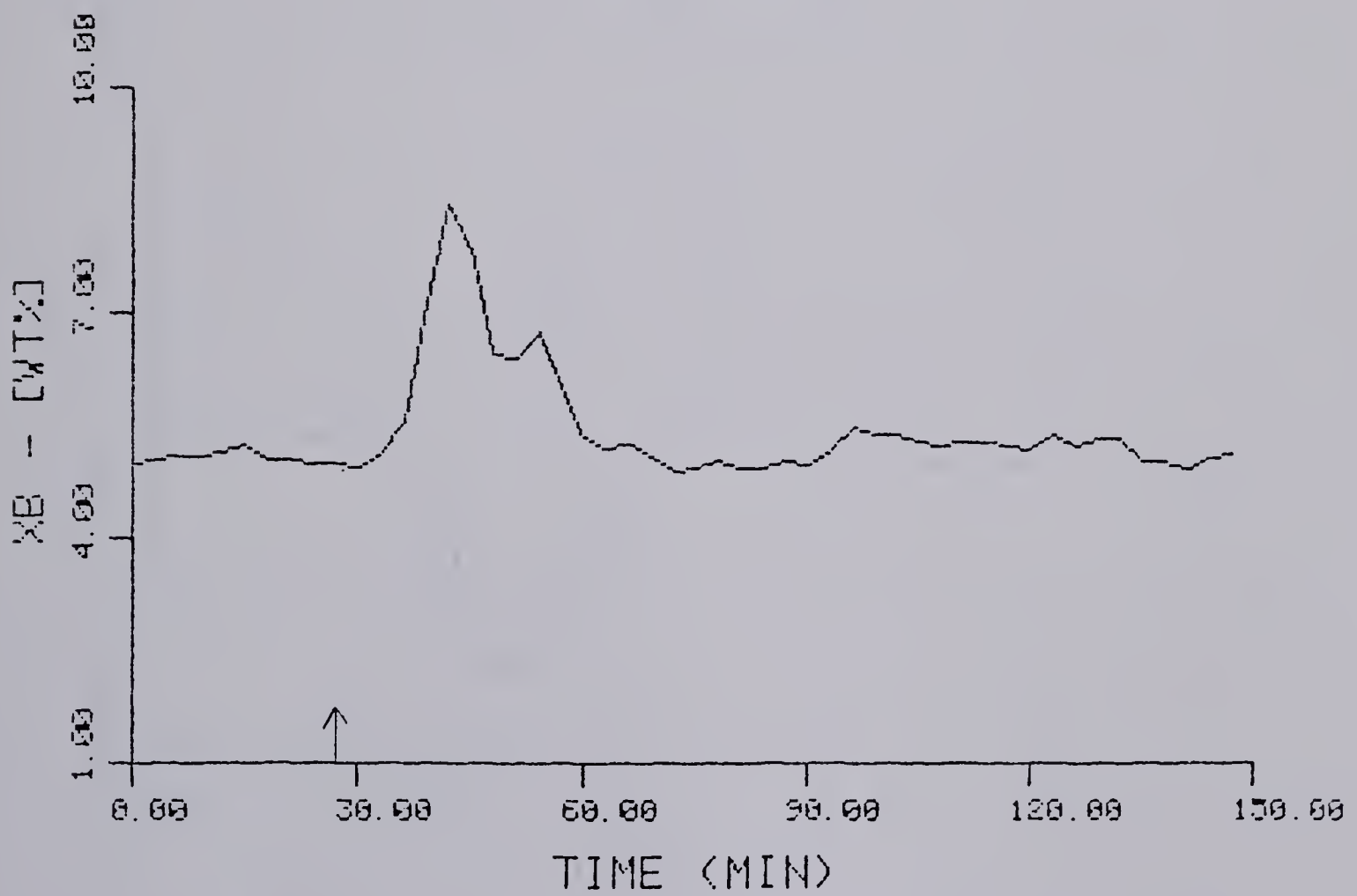
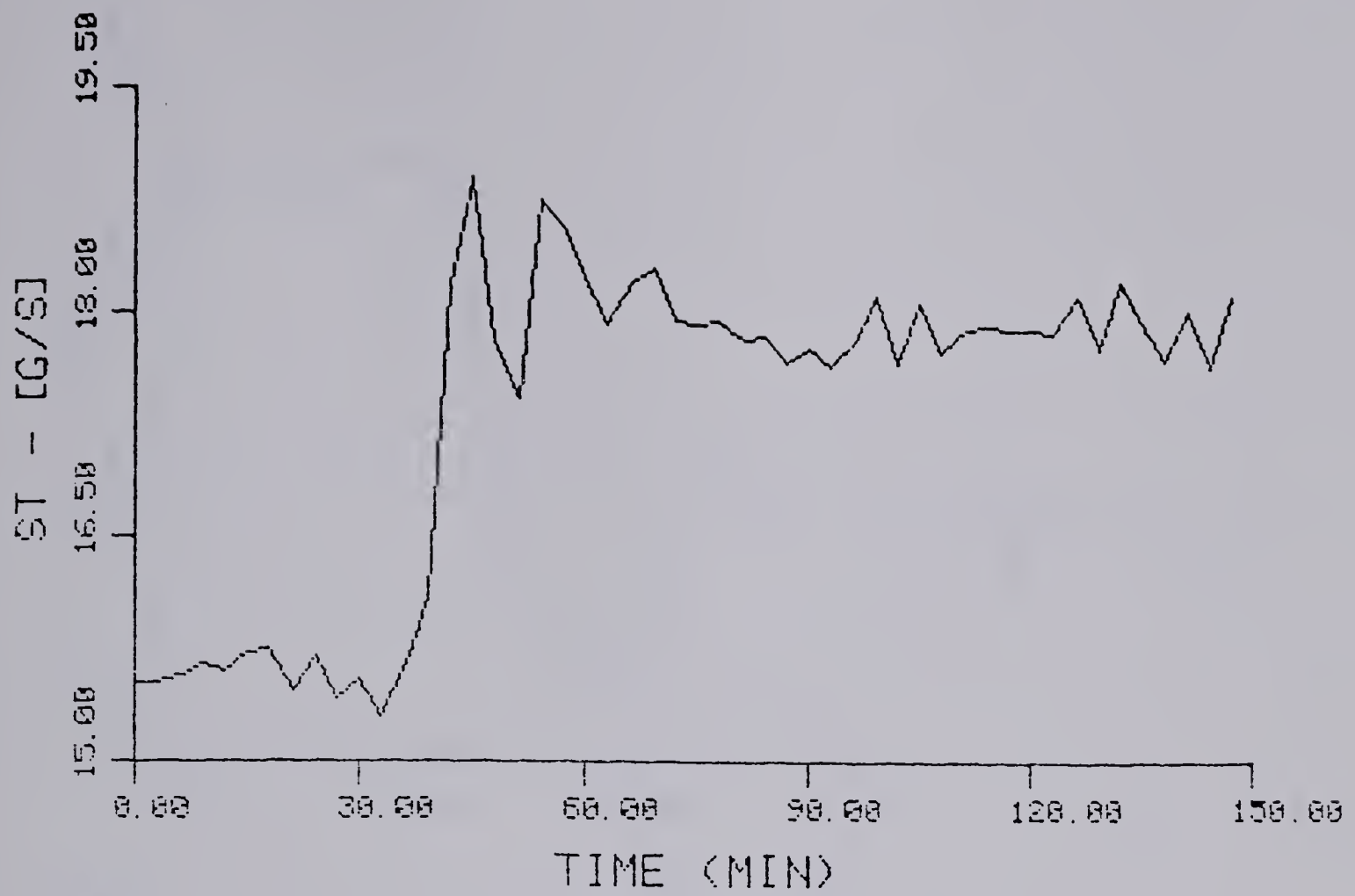


Figure 5.29 ST Control of Bottom Composition for a +25% Step Change in Feed Flow Rate With Q-WT ($k = 0.90$ and $k = 0.47$)

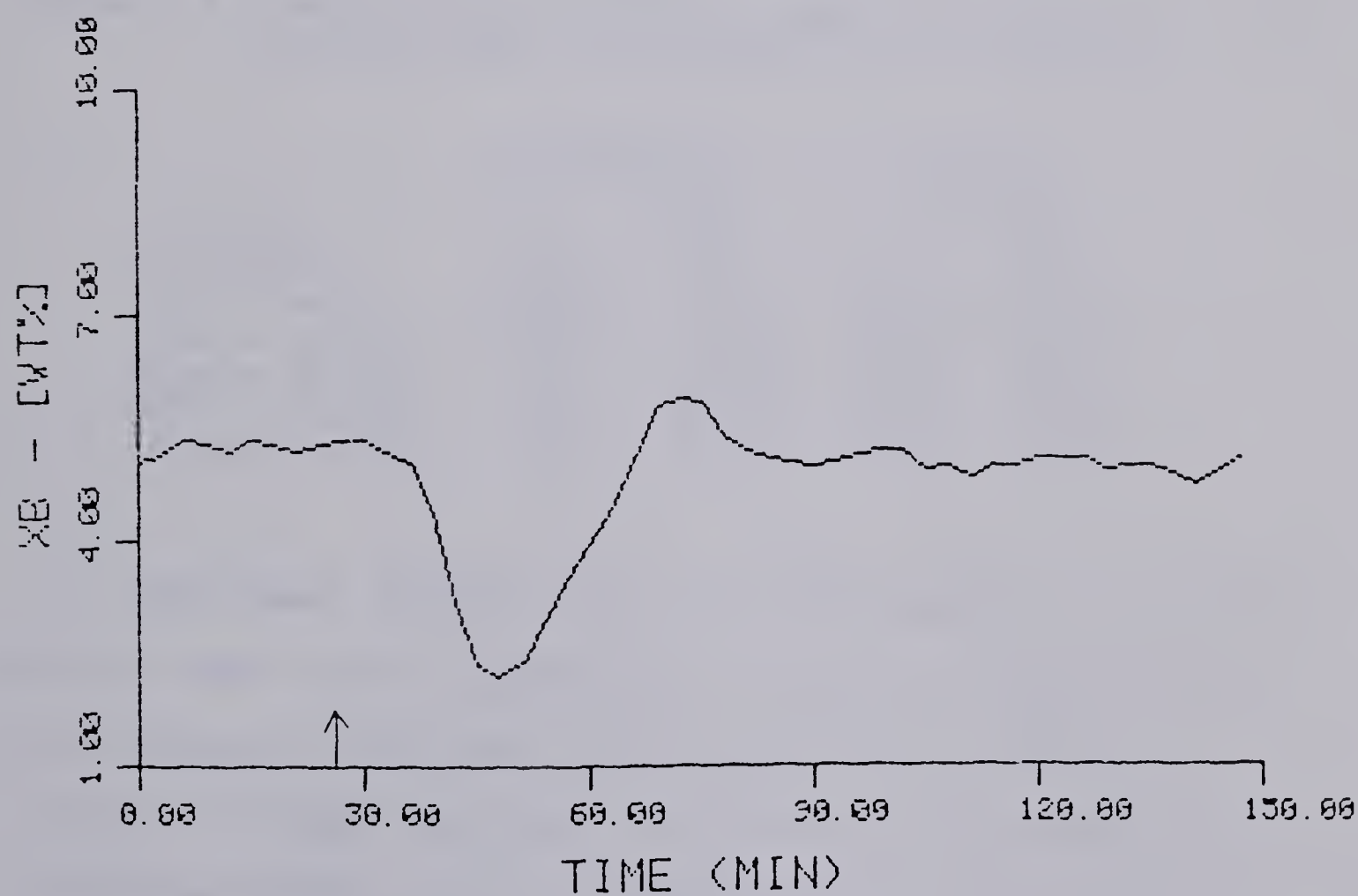
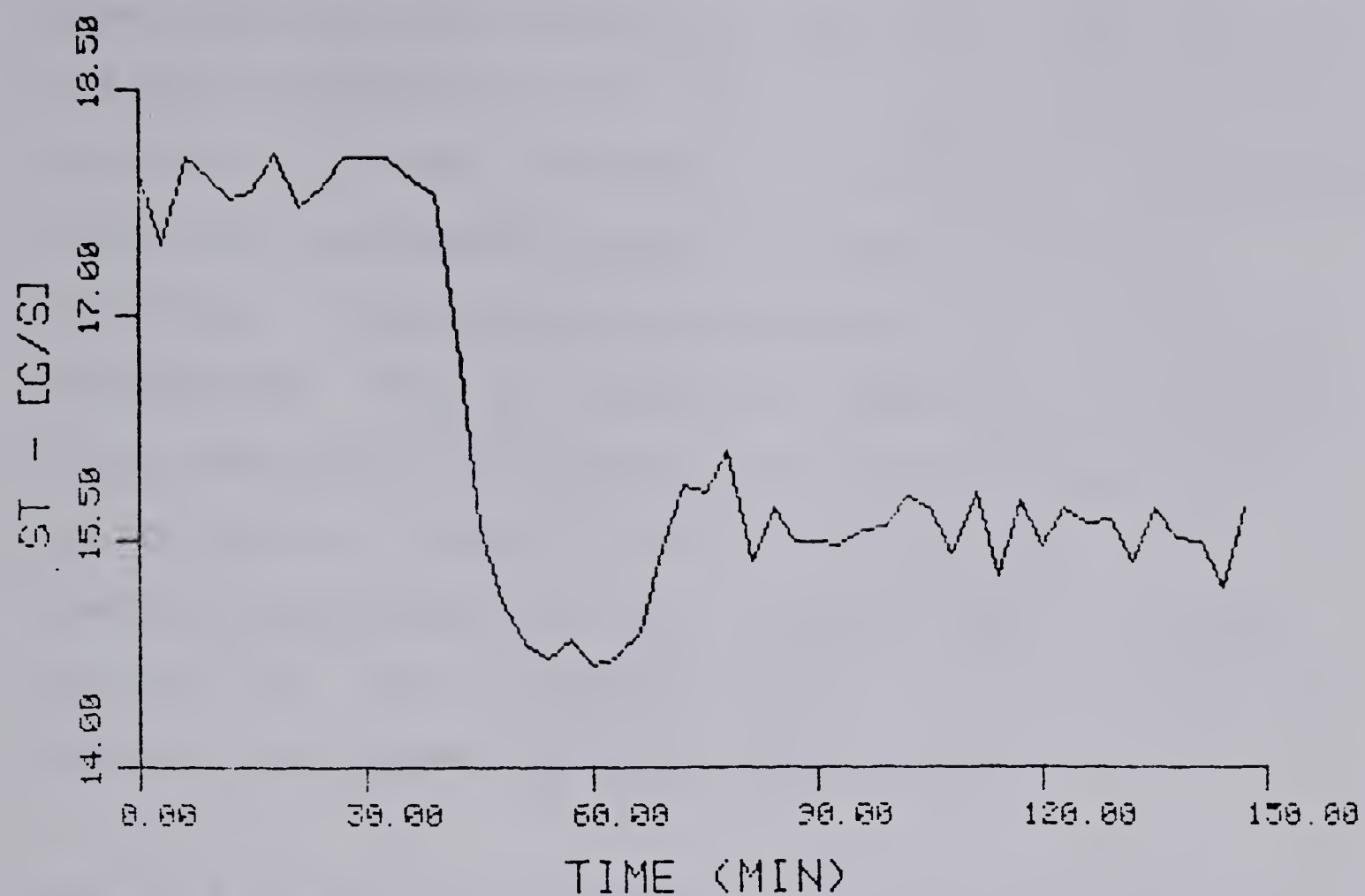


Figure 5.30 ST Control of Bottom Composition for a Step Decrease in Feed Flow Rate to its Normal Steady State Value With Q-WT ($k_p = 0.90$ and $k_i = 0.47$)

compensator than the Q-weighting constants. Also, the final Q-weighting parameters are closer to the original estimates. In addition, the PID constants for the different disturbances differ by a considerable amount in comparison to the variations in the Q-weighting parameters for the different disturbances. This can be seen by comparing the difference in the two sets of PID constants for bottom composition control given in Table 5.1 with the difference in the Q-weighting parameters for the responses shown in Figure 5.27 to 5.30. This is because the STC is able to adapt to the nonlinear dynamic behavior of the column.

Table 5.7 Comparison of Bottom Composition Absolute Error Values (over 1 1/2 hour) for PID and STC

Type of Disturbance	PID Control		ST Control	
	Figure	SAE (wt%)	Figure	SAE (wt%)
-25% to Feed	5.7	18.43	5.27	19.70
Return to S.S.	5.8	14.00	5.28	14.90
+25% to Feed	5.5	20.72	5.29	18.76
Return to S.S.	5.6	26.04	5.30	20.51

Additional evidence that the STC handles nonlinearity better than the PID compensator is seen by comparing the performances of the two controllers for the +25% step increase in feed flow and the subsequent step decrease to the normal feed rate (cf. Figures 5.6-5.7 and 5.29-5.30). From the SAE values given in Table 5.7, it can be seen that the STC, compared to the PID compensator, provided similar

control performance for these different step changes in feed flow.

The effect of the addition of feedforward (FF) control action to the STC was studied using the same controller parameters and options given in Table 5.6 except that the values of the G1 and F parameters are initially set equal to the final values obtained when using the STC without feedforward action.

The bottom composition responses for the same four different feed flow rate disturbances are given in Figures 5.31 to 5.34 with the absolute error values for these responses and those of the previous PID and STC tests summarized in Table 5.8.

Table 5.8 Comparison of Bottom Composition Absolute Error Values (over 1 1/2 hour) for PID, STC and STC+FF

Type of Disturbance	PID Control		ST Control		STC + FF	
	Figure	SAE (wt%)	Figure	SAE (wt%)	Figure	SAE (wt%)
-25% to Feed	5.7	18.43	5.27	19.70	5.31	26.27
Return to S.S.	5.8	14.00	5.28	14.90	5.32	9.93
+25% to Feed	5.5	20.72	5.29	18.76	5.33	9.17
Return to S.S.	5.6	26.04	5.30	20.51	5.34	8.41

The results given are for STC runs that already have 'good' parameter estimates. Except for the -25% step decrease in feed flow rate from its steady state value, the STC with feedforward control action improves the bottom

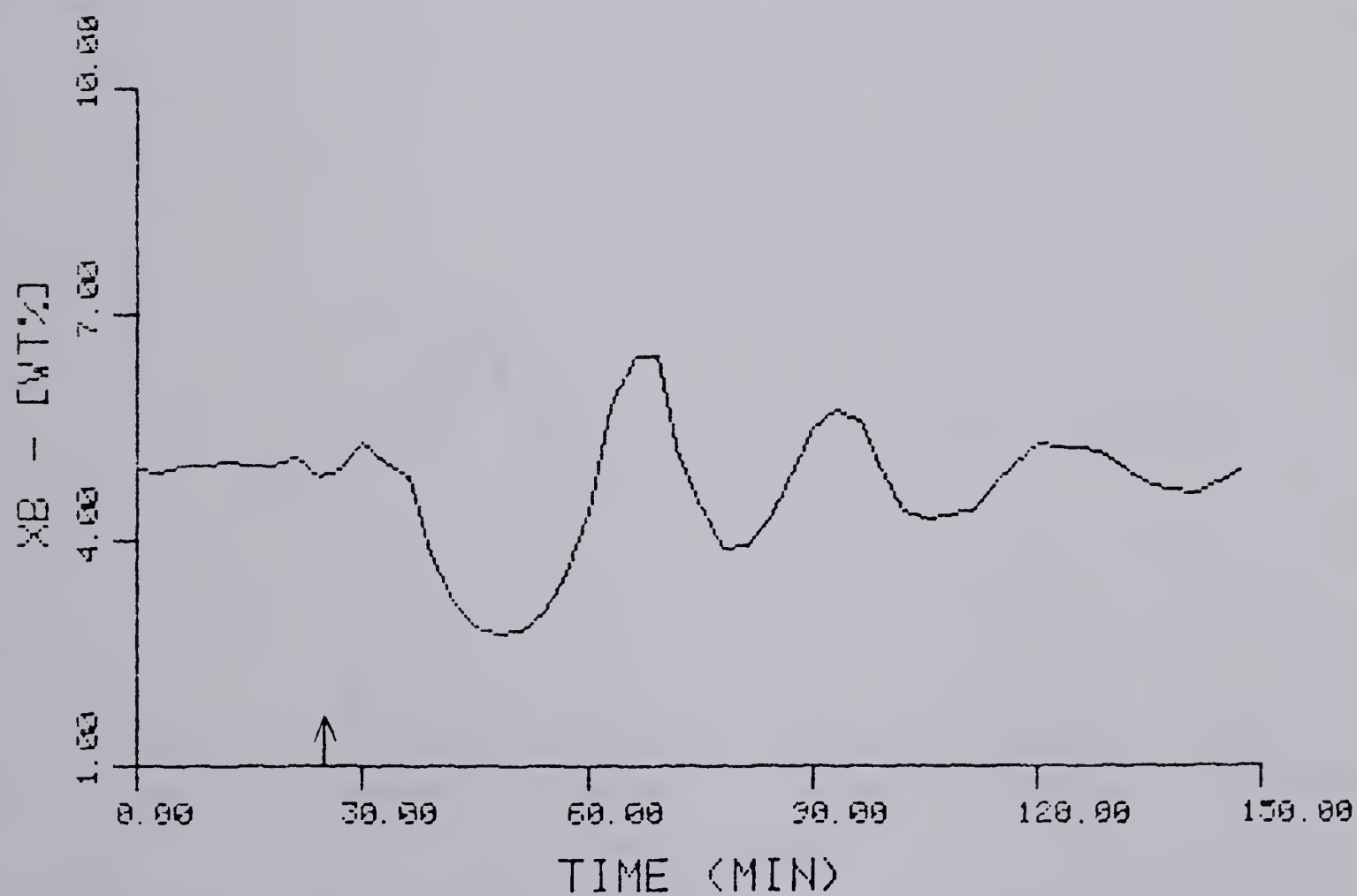
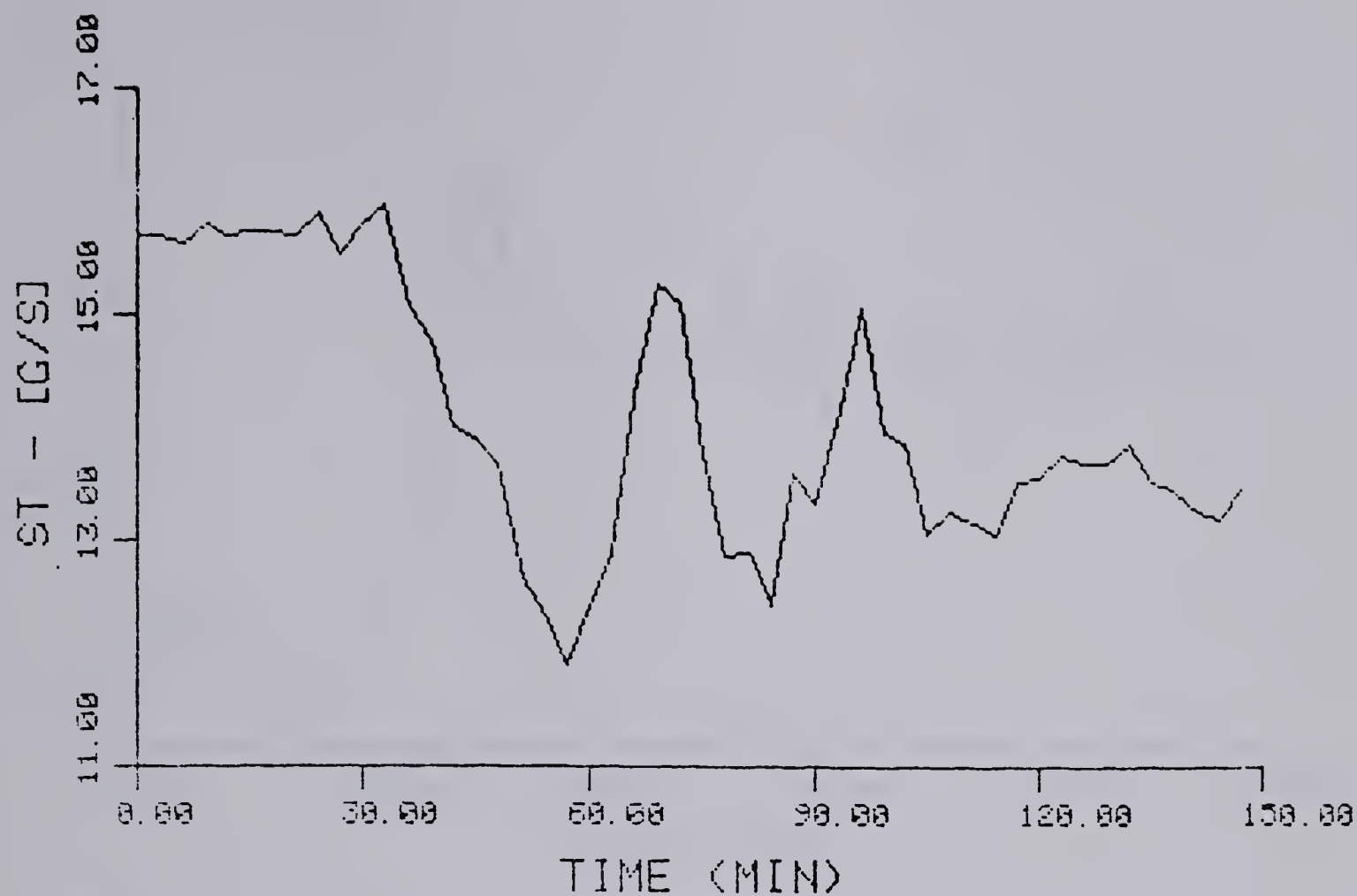


Figure 5.31 STC + FF Control of Bottom Composition for a -25% Step Change in Feed Flow Rate With Q-WT ($k_p = 0.90$ and $k_i = 0.47$)

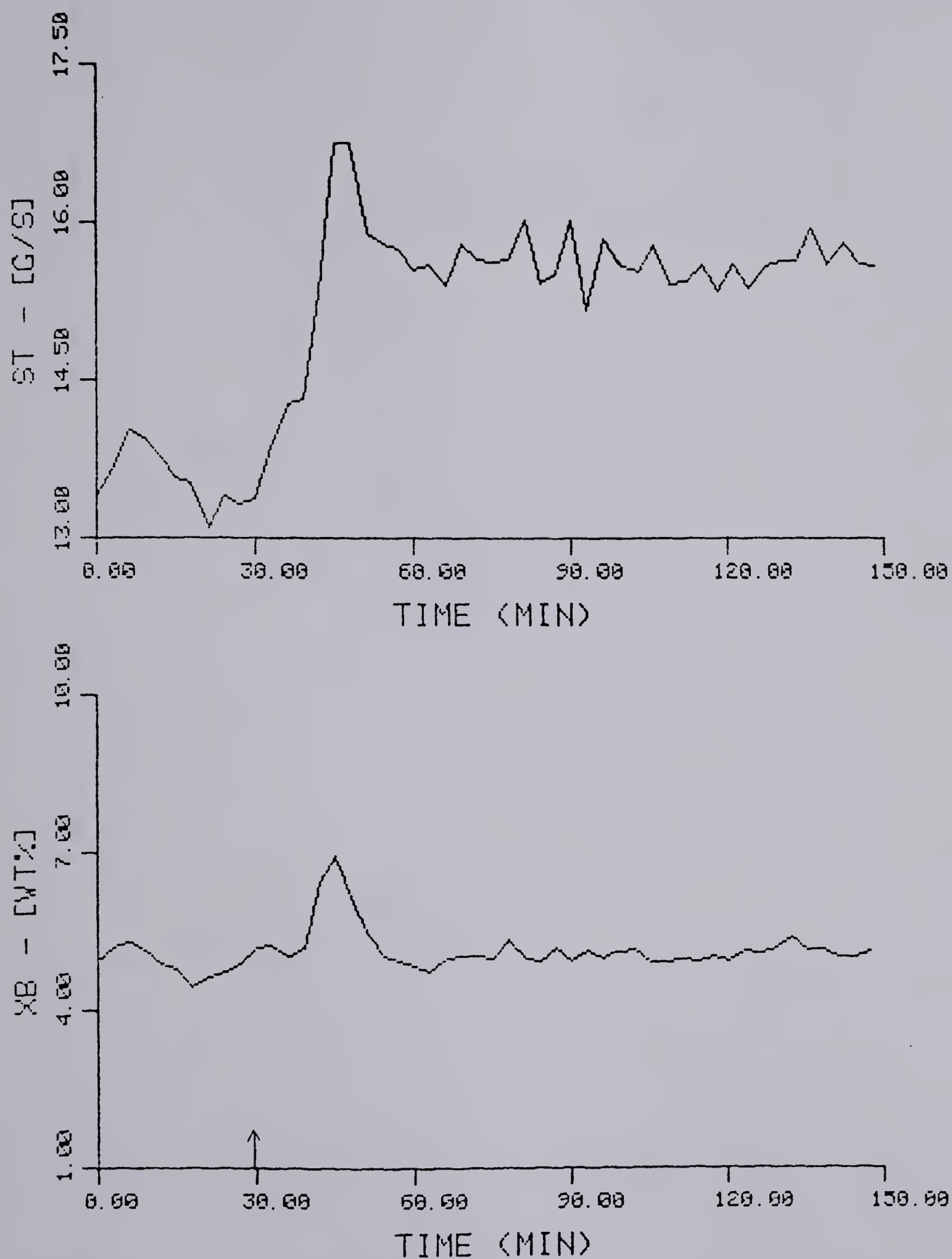


Figure 5.32 STC + FF Control of Bottom Composition for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT ($k_p = 0.90$ and $k_i = 0.47$)

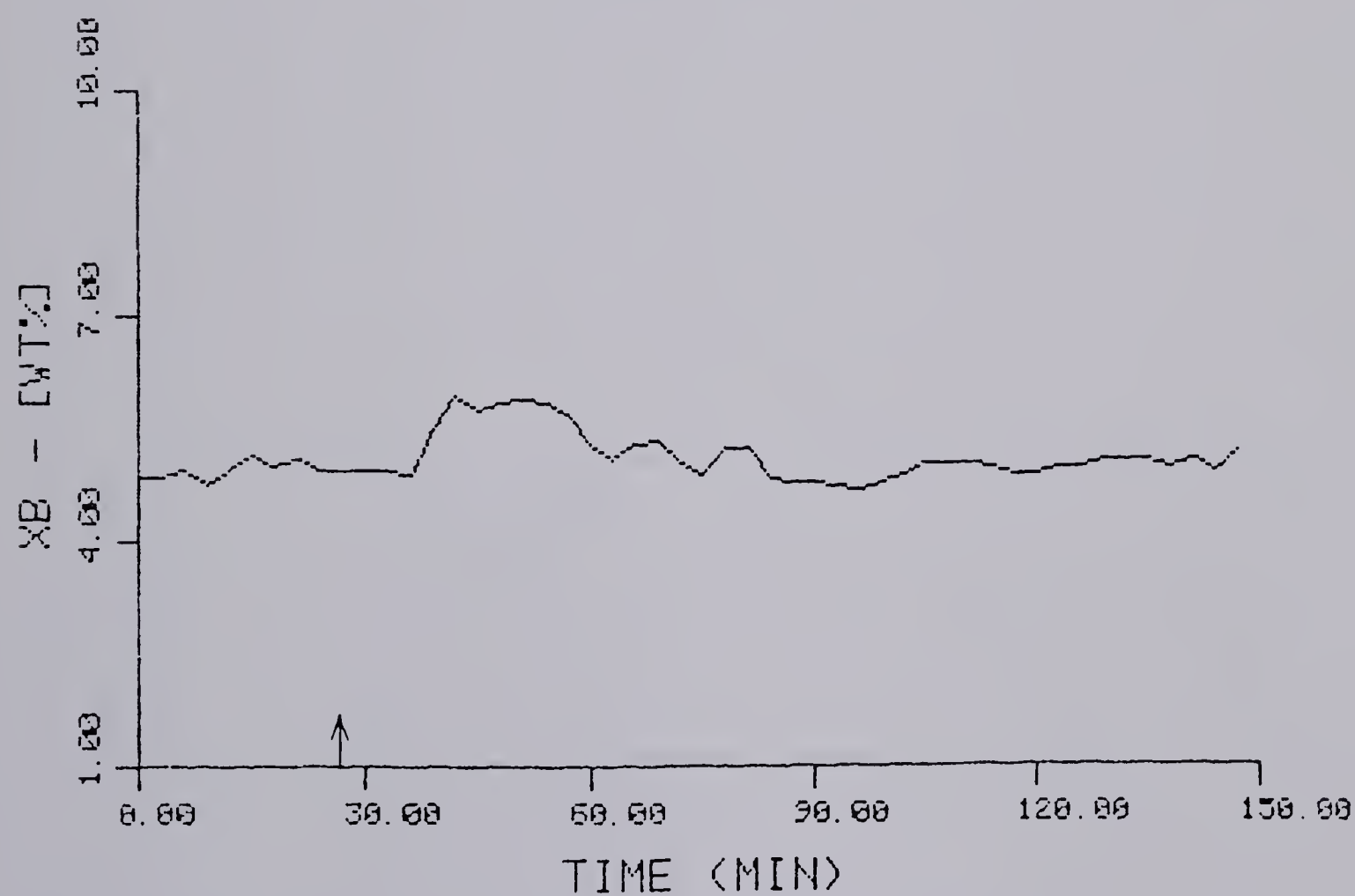
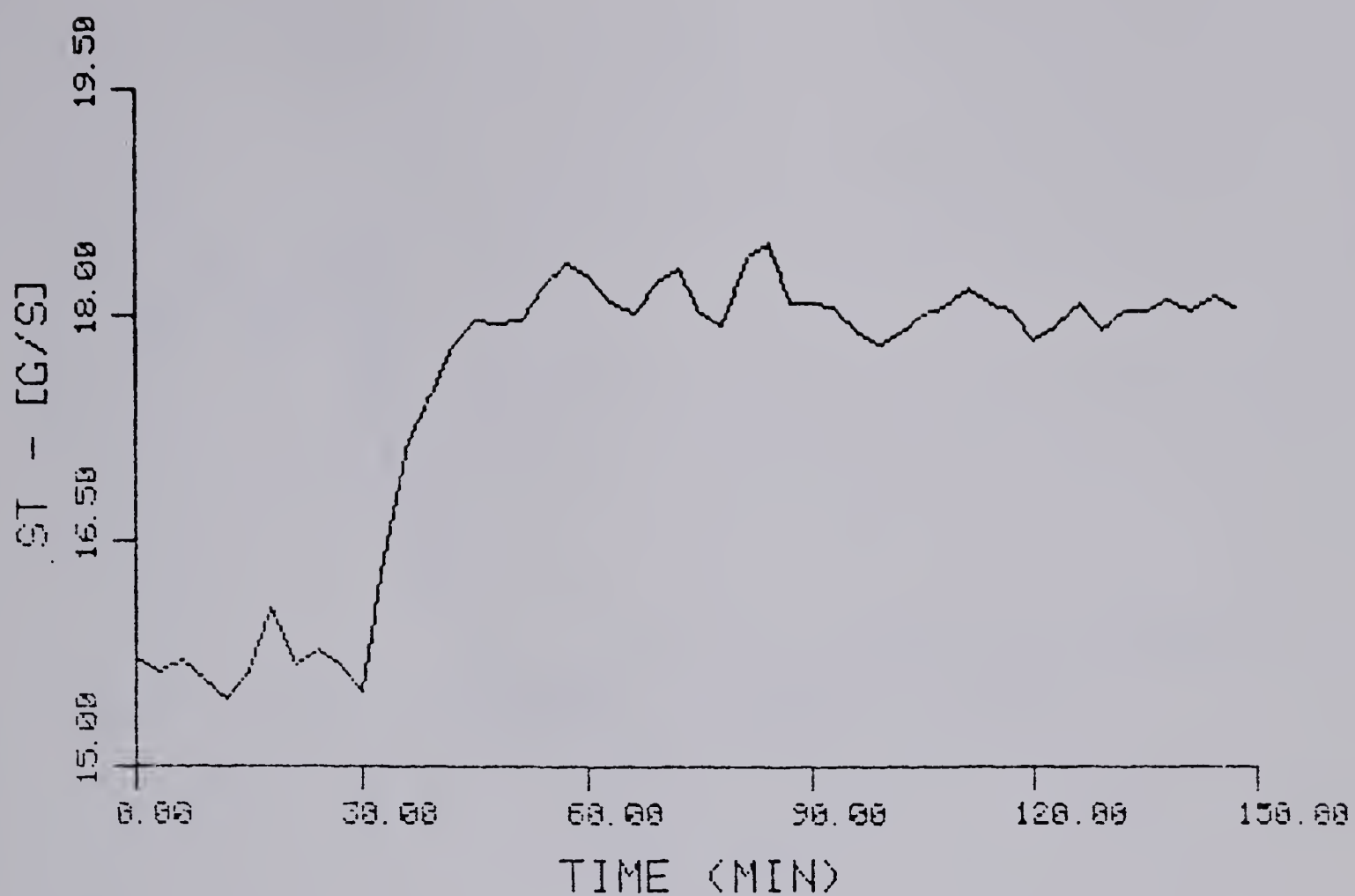


Figure 5.33 STC + FF Control of Bottom Composition for a +25% Step Change in Feed Flow Rate With Q-WT ($k_p = 0.90$ and $k_i = 0.47$)

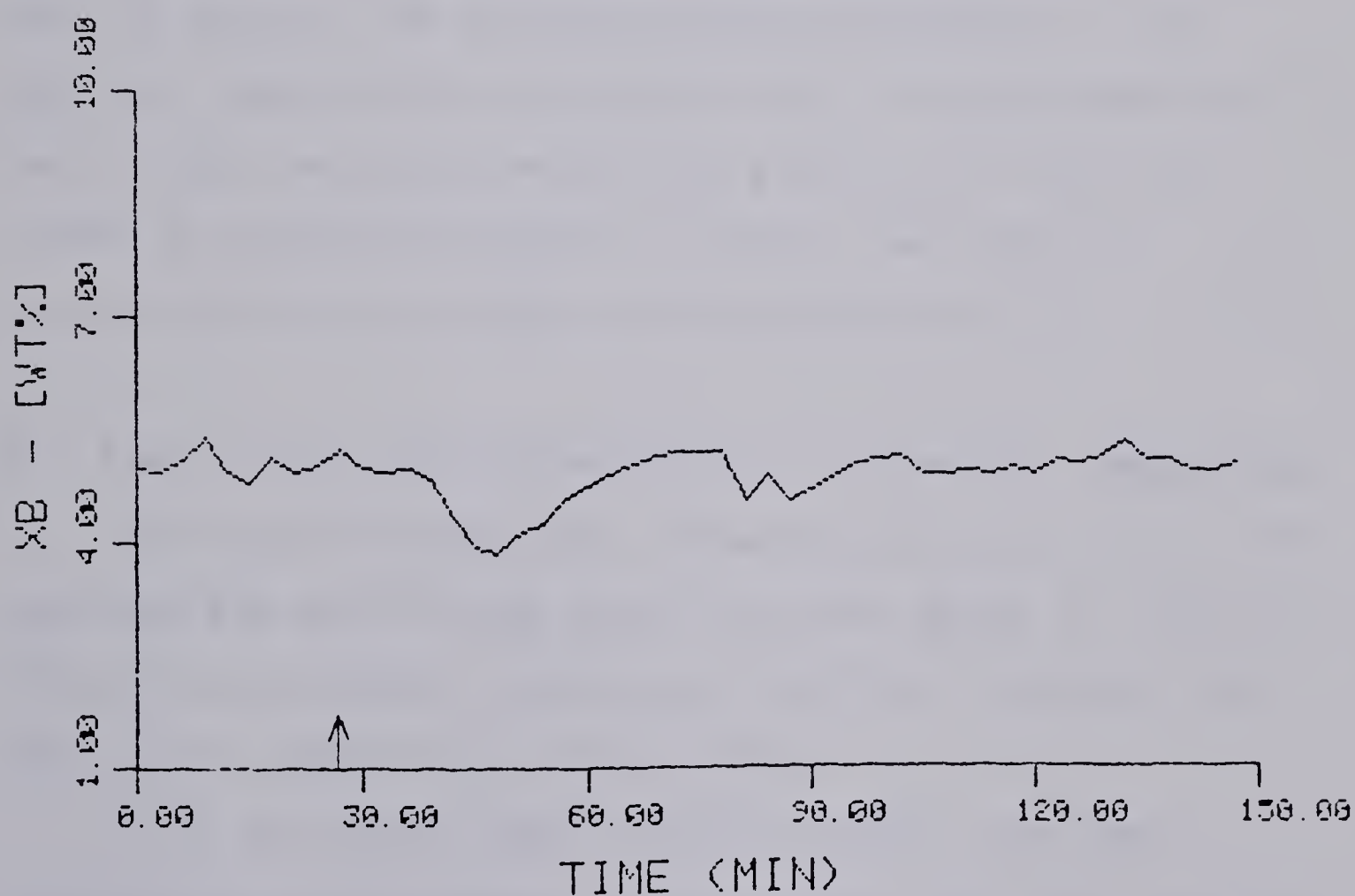
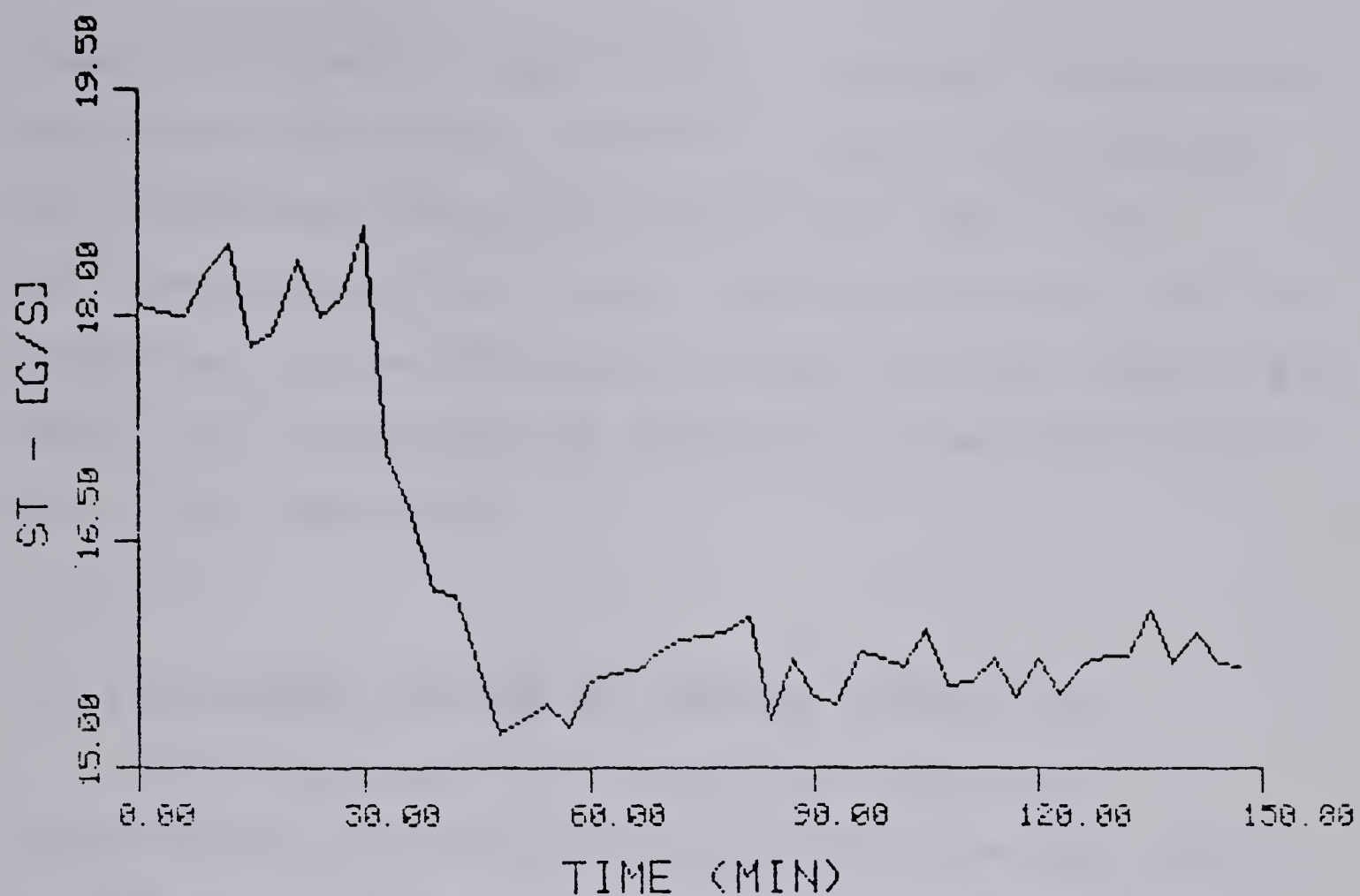


Figure 5.34 STC + FF Control of Bottom Composition for a Step Decrease in Feed Flow Rate to its Normal Steady State Value With Q-WT ($k_p = 0.90$ and $k_i = 0.47$)

composition control significantly. The error values using feedforward are almost one-half of those values obtained when feedforward control action was not used. The self-tuning controller did not perform better than the PID controller for the -25% step decrease from the steady state feed flow rate because the feedforward terms have not had time to be identified.

5.7 SIMULTANEOUS CONTROL OF TERMINAL COMPOSITION

Simultaneous control of overhead and bottom compositions using the self-tuning controller was carried out for each of the different feed disturbances. The terminal compositions were controlled using two separate STC (multiloop configuration, ML-STC) and two STC that were linked by feedforward terms to cancel the effect of interaction (multivariable configuration, MV-STC).

5.7.1 Multiloop Self-Tuning Control of Terminal Composition

The original controller parameters and options for the two separate self-tuning controllers are given in Table 5.9. The initial parameters values are the final estimates from the single composition control tests.

After an initial identification period, the feed disturbance is introduced. It was found that the top loop Q-weighting had to be de-tuned to prevent oscillatory behavior. The composition behavior, for the de-tuned Q-weighting test for the -25% step decrease from the steady

Table 5.9 Controller Parameters and Options for Multiloop and Multivariable STC

Identification Scheme	:	Recursive Square root (top and bottom)			
G Polynomial	:	EB (top and bottom)			
Number of Parameters	:	Top		Bottom	
	:	NG1 = 3		NG1 = 4	
	:	NF = 2		NF = 3	
	:	ND = 0		ND = 5 (FF added)	
	:	NH = 1		NH = 1	
	:	NG2 = 2 (MV-STC)		NG2 = 3 (MV-STC)	
	:	K1 = 1		K1 = 2	
	:	K2 = 0 (MV-STC)		K2 = 0 (MV-STC)	
	:	K3 = 0		K3 = 1 (FF added)	
Parameter fixed	:	$h_0 = -1$ (top and bottom)			
Initial Value of Parameters	:	Top		Bottom	
	:	G1	F	G1	F D
	:	0.2090	0.7382	0.1586	1.0340 0.03257
	:	-.0409	0.1161	0.0548	0.1695 0.04689
	:	-.0834		-.0997	-.6200 0.00937
	:			-.0763	-.03920
	:				-.03590
Q-weighting	:	Top		Bottom	
	:	$k_p =$	3.676	1.145	(g/s)/wt%
	:	$k_i =$	0.919	0.226	(g/s)/wt%
P-weighting	:	Set to 1 (top and bottom)			
Sampling Interval	:	3 minute (top and bottom)			
Forgetting Factor	:	0.99 (top and bottom)			
Initial \underline{P}^C Matrix	:	0.01 \underline{I} (top and bottom)			
Incremental Control Limit	:	Top		Bottom	
	:	2.30 g/s		2.25 g/s	

state feed flow rate and the subsequent return to the normal value are shown in Figures 5.35 and 5.36. The control performance that was obtained with feedforward control action added to the bottom STC is shown in Figures 5.37 and 5.38. A summary of the absolute error values over a period of 30 samples after the feed flow rate disturbance was introduced is given in Table 5.10.

Table 5.10 Summary of Top and Bottom Composition Absolute Error Values for Multiloop STC With and Without Feedforward (FF) added

Type of Control		T y p e O f D i s t u r b a n c e					
		-25% to Feed		Return to S.S.		+25% to Feed	
		Figure	SAE (wt%)	Figure	SAE (wt%)	Figure	SAE (wt%)
ML - STC	top	5.35	2.32	5.36	2.17		
	bot	5.35	23.75	5.36	19.71		
ML - STC + FF	top			5.37	3.74	5.38	2.27
	bot			5.37	14.59	5.38	8.97

The SAE values show that the control performance of the multiloop self-tuning controller was comparable to that of the well tuned PID compensator. It should be noted that as with bottom composition control, implementation of the STC required much less tuning than was the case for the PID regulator. In fact, the Q-weighting for the bottom composition STC was not re-tuned, from the single loop values, for use with multiloop control. In addition, proportional, integral and derivative actions are involved in the PID compensator while only proportional and integral

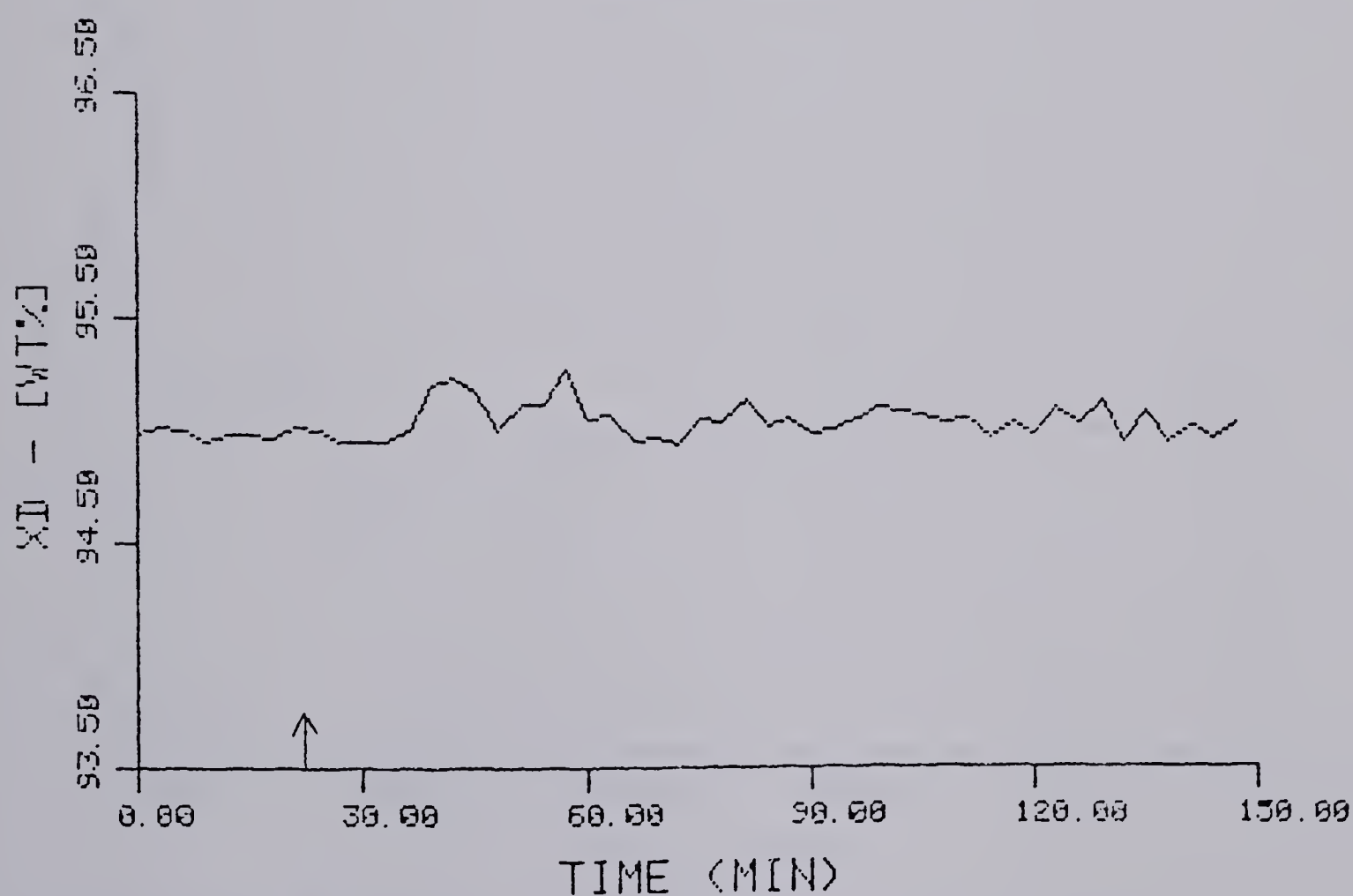
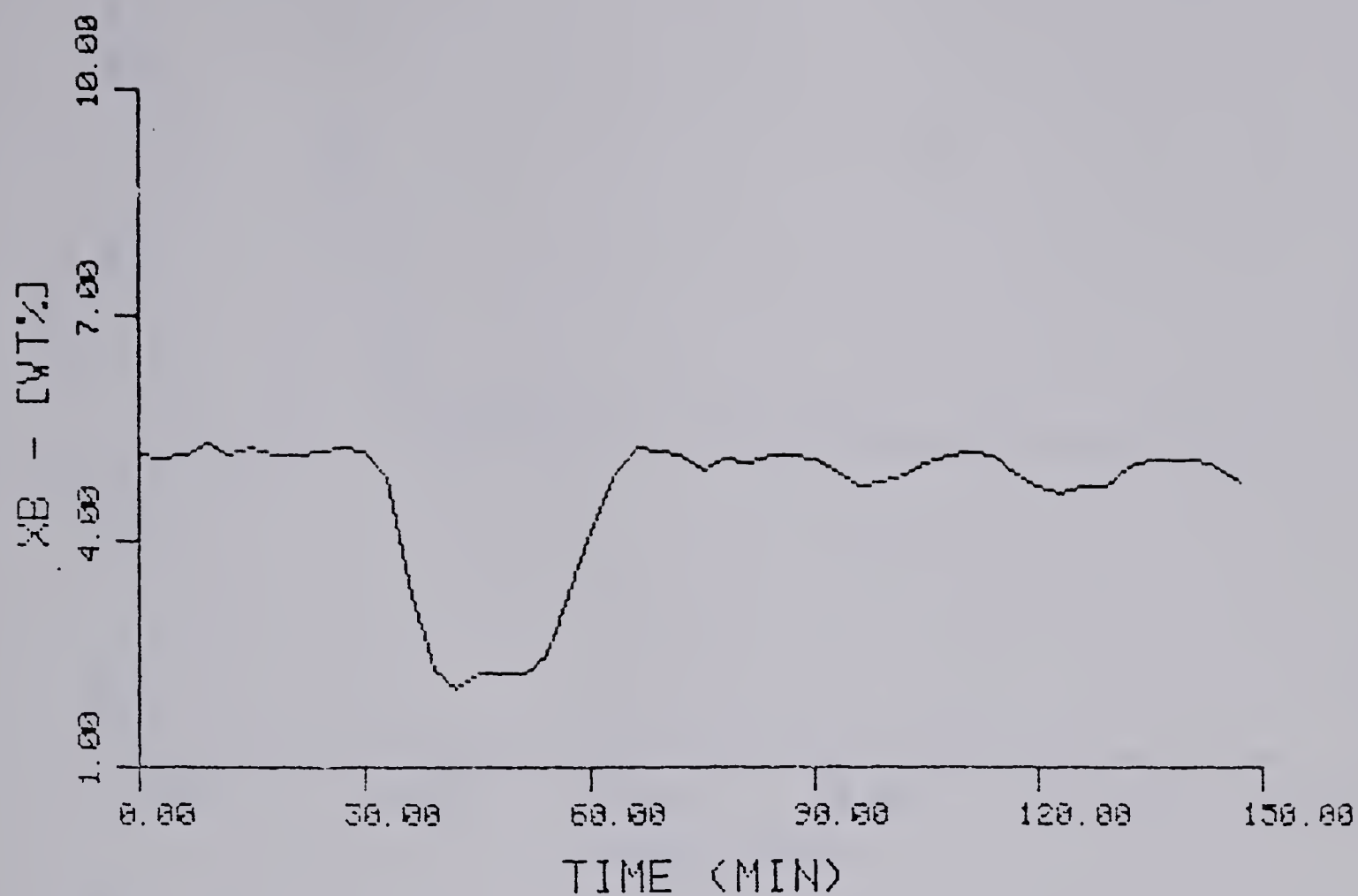


Figure 5.35 ML-ST Control of Terminal Compositions for a -25% Step Change in Feed Flow Rate With Q-WT (top: $k_p = 2.7$ $k_i = .45$ bot: $k_p = .90$ $k_i = .47$)

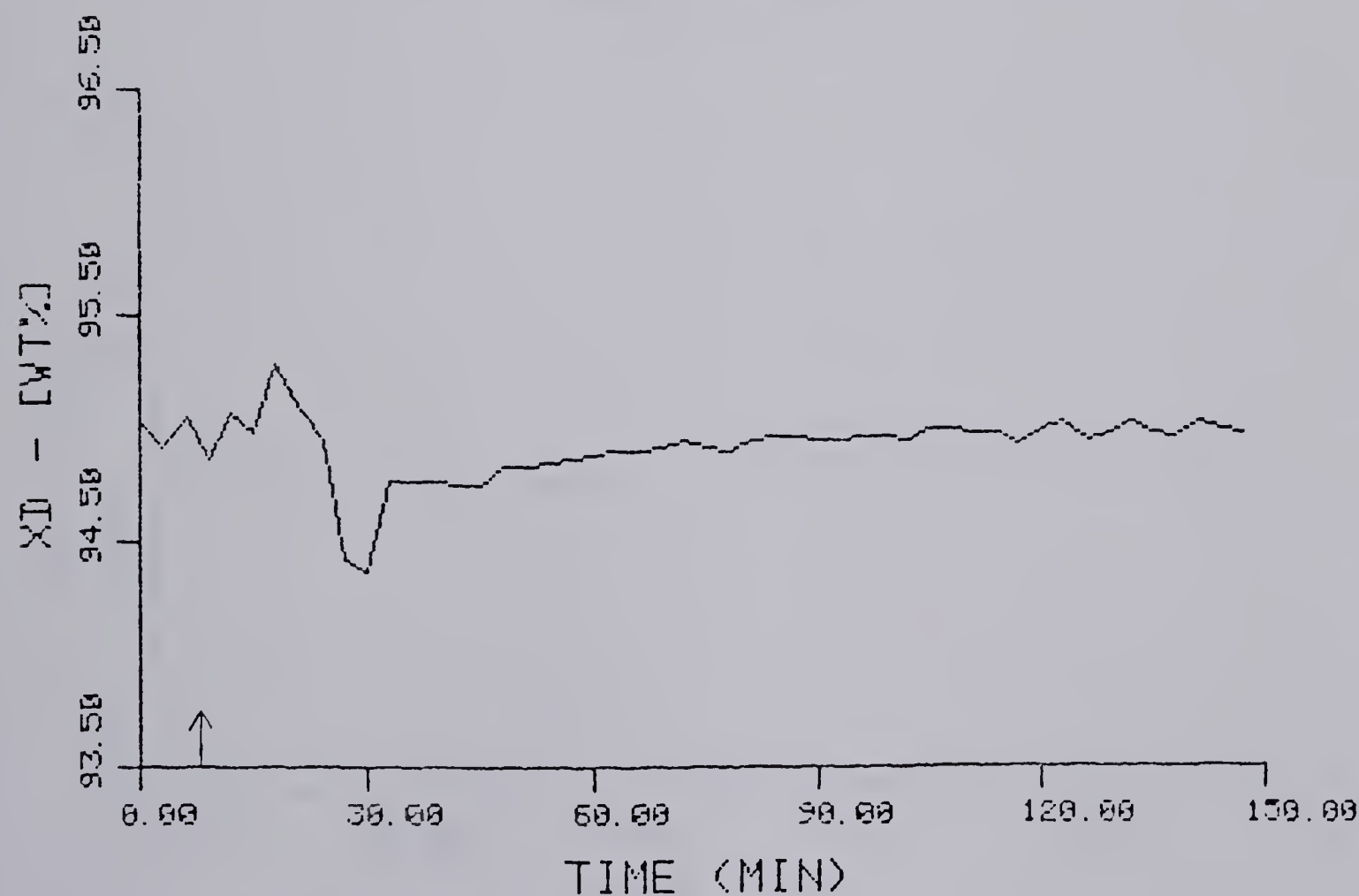
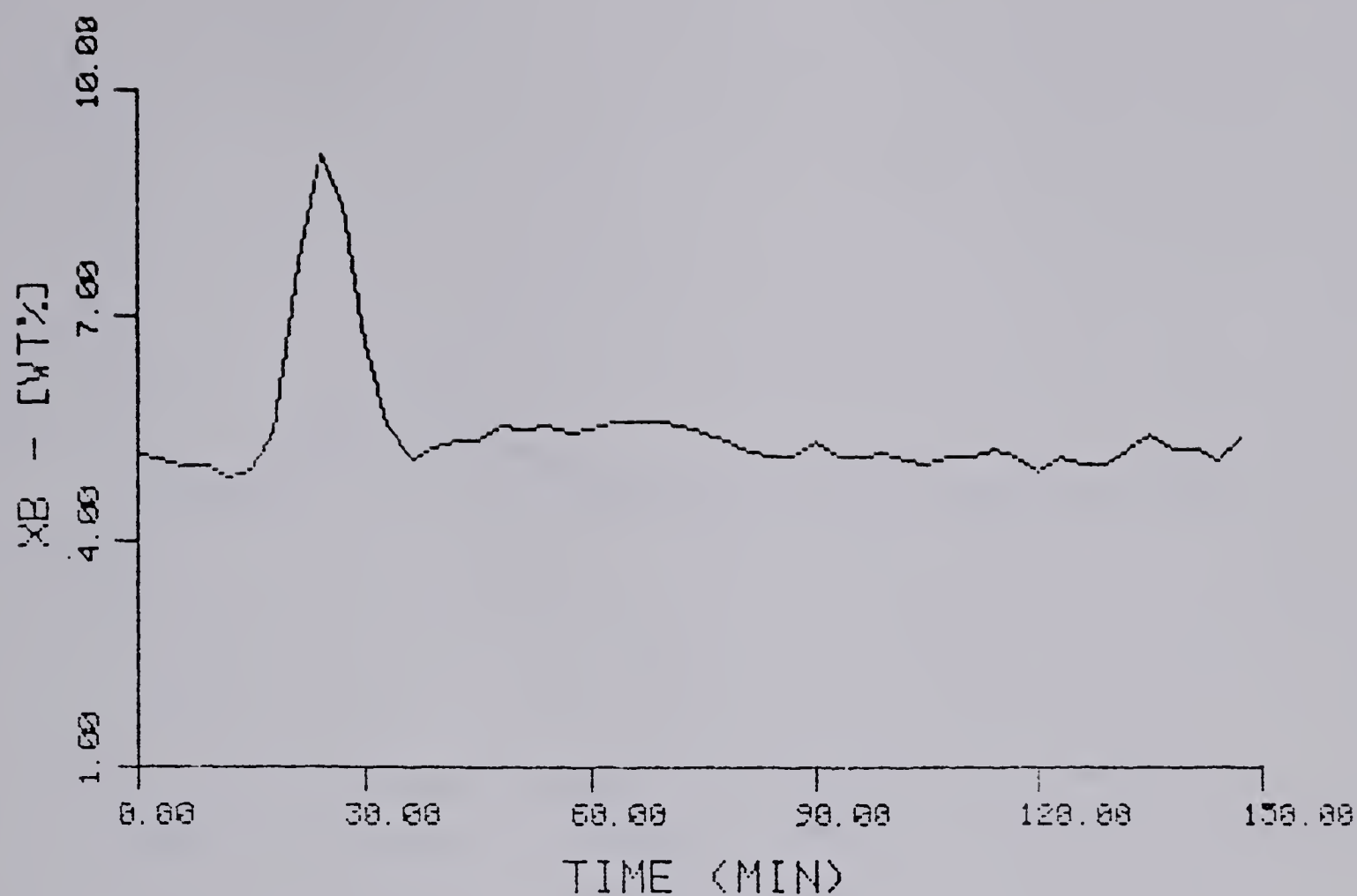


Figure 5.36 ML-ST Control of Terminal Compositions for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT (top: $k_p = 2.7$ $k_i = .45$ bot: $k_p = .90$ $k_i = .47$)

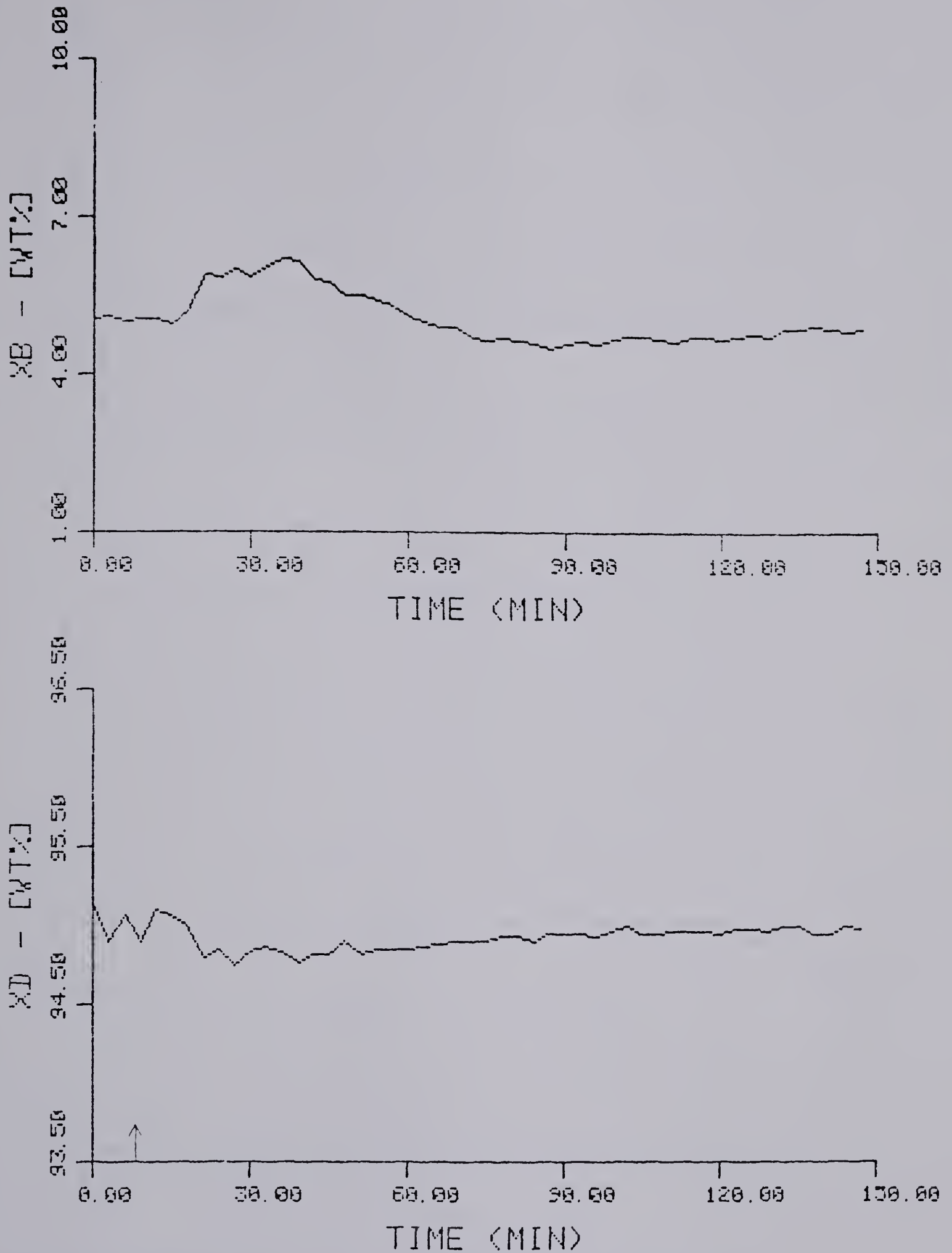


Figure 5.37 ML-STC + FF (bottom) Control of Terminal Compositions for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT (top: $k_p = 2.7$ $k_i = .45$ bot: $k_p = .90$ $k_i = .47$)

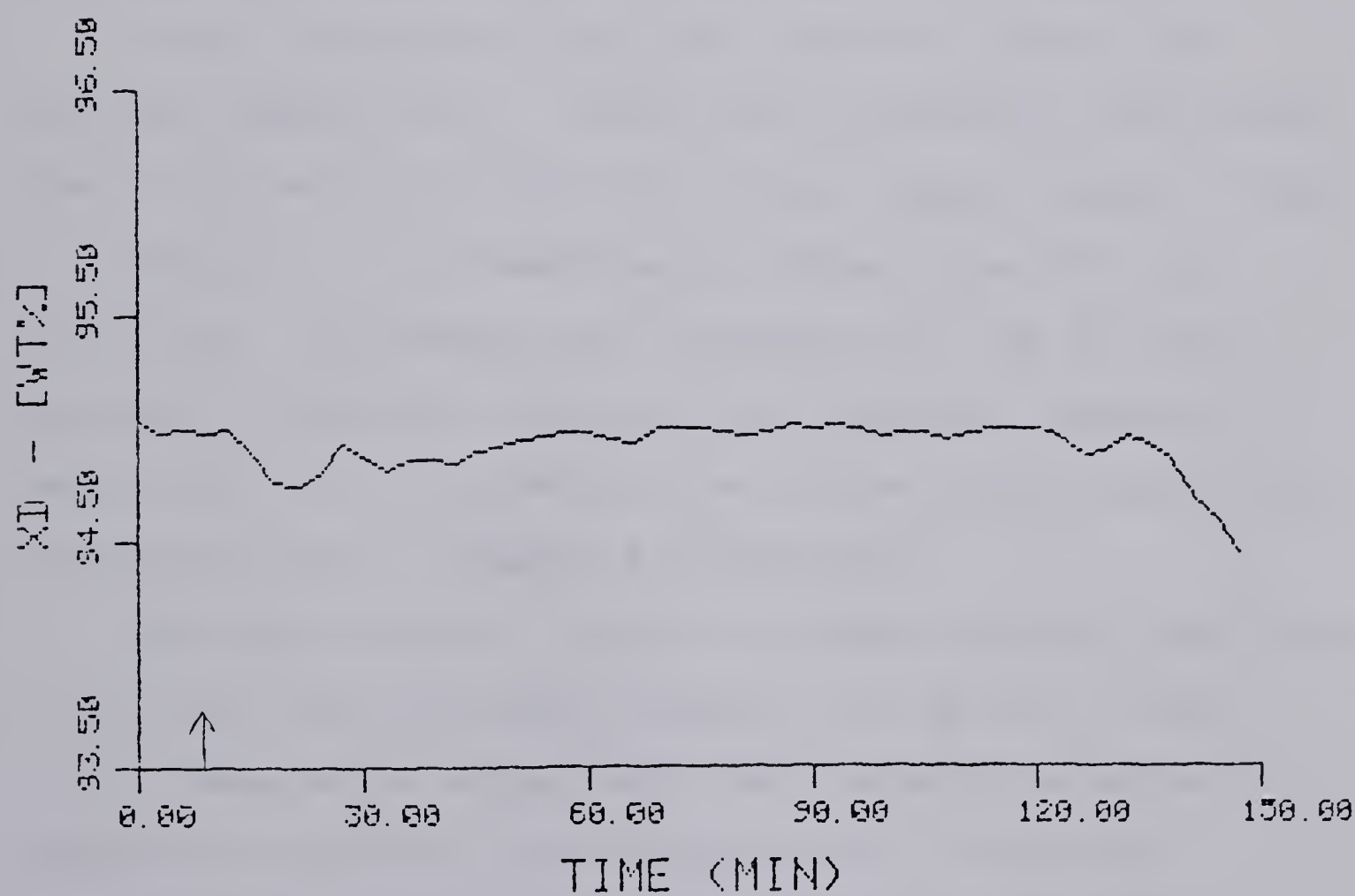
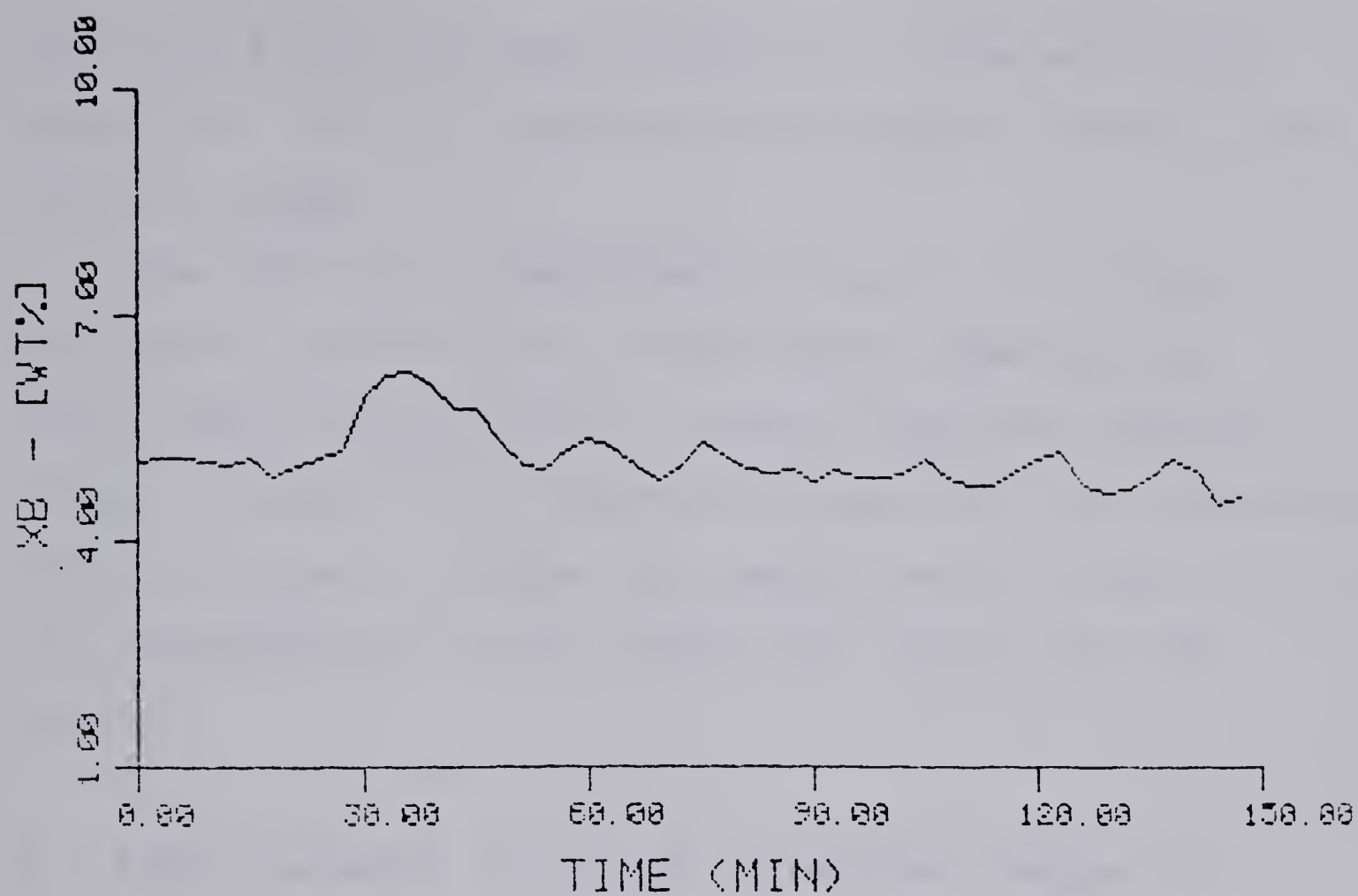


Figure 5.38 ML-STC + FF (bottom) Control of Terminal Compositions for a +25% Step Change in Feed Flow Rate With Q-WT (top: $k_p = 2.7$ $k_i = .45$ bot: $k_p = .90$ $k_i = .47$)

actions are used with the Q-weighting of the self-tuning controller. The STC response should improve if derivative action is added.

The addition of feedforward action for the bottom self-tuning controller has significantly improved the control performance as can be seen by examining the SAE values in Table 5.10. The bottom composition SAE values are greatly reduced while the top composition SAE values for the STC are comparable to the values that result with PID control.

5.7.2 Multivariable ST-Control of Terminal Composition

A multivariable STC was also used to control the terminal compositions. The original parameters and options used by the multivariable STC are the same as those in Table 5.8 except that the feedforward linkage terms (G_2) are identified. The composition responses for the 25% step decrease in feed flow rate and the subsequent increase in feed flow rate to return the flow to the normal steady state value are given in Figures 5.39 and 5.40.

The absolute error values for these responses and those of the multiloop STC and PID control are given in Table 5.11. These error values show that the multivariable STC performed as well as the PID compensator for bottom composition control, but provided significantly better control of top composition control than did PID control. The top composition error values with multivariable STC are

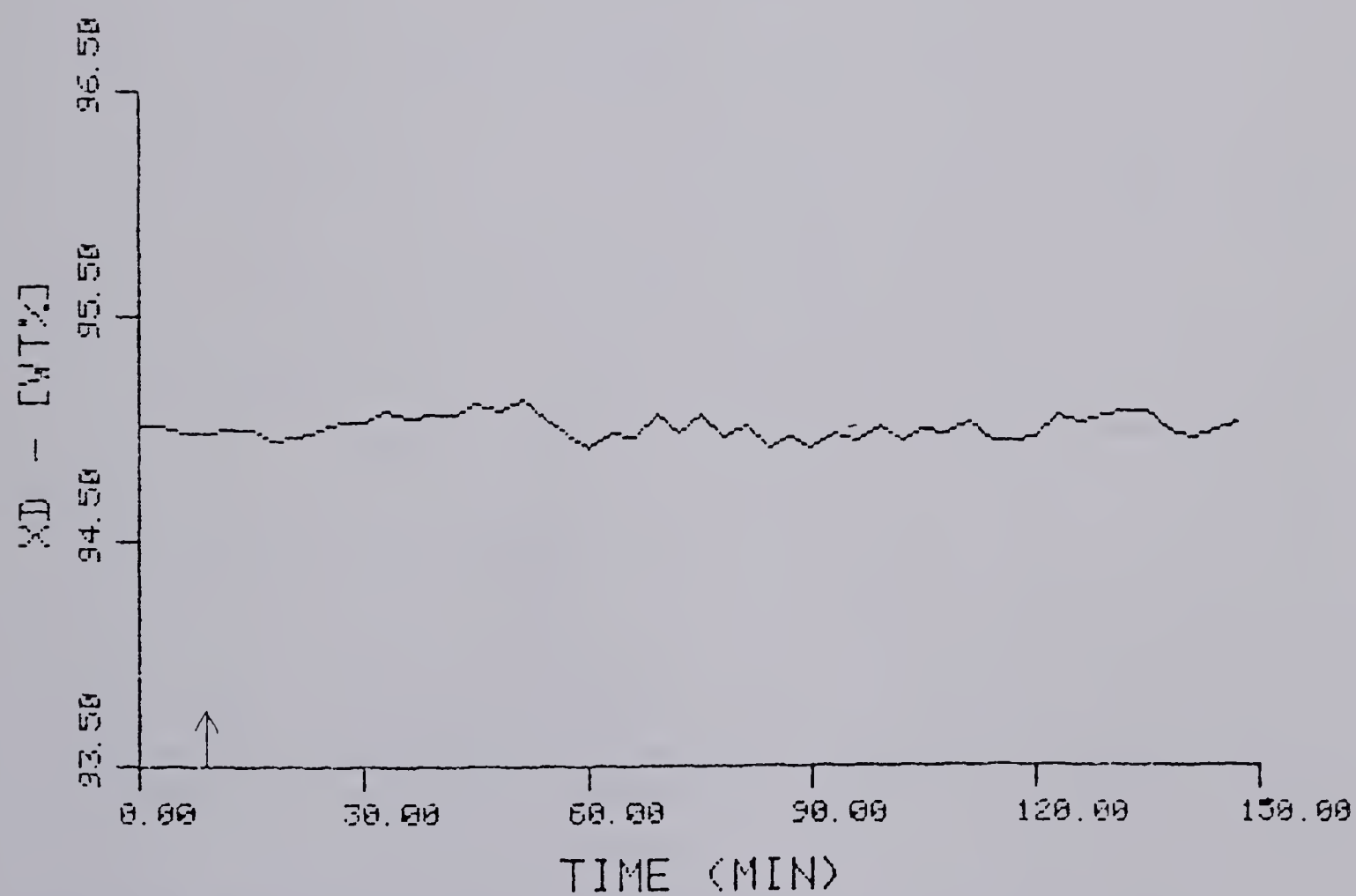
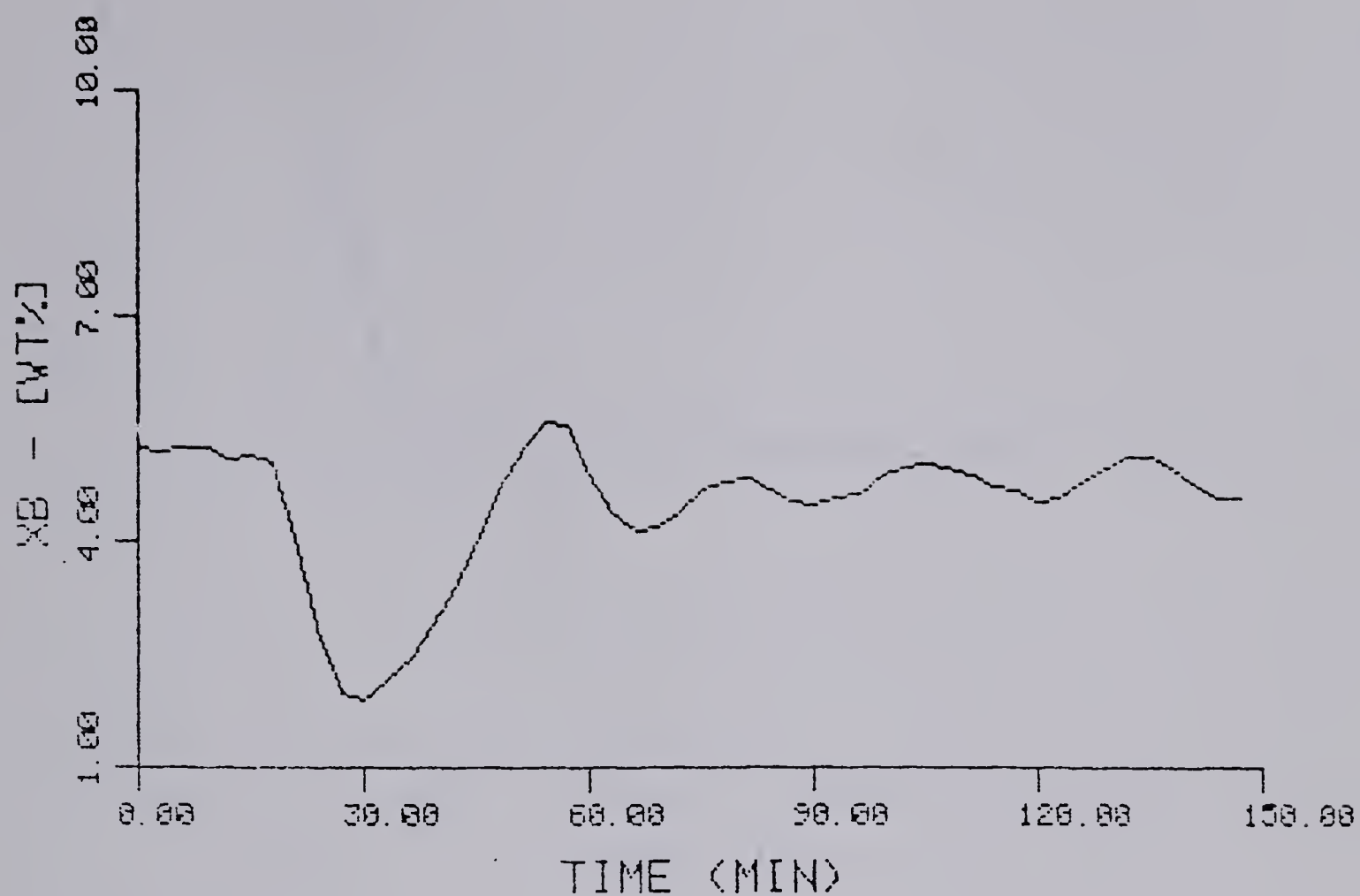


Figure 5.39 MV-ST Control of Terminal Compositions for a -25% Step Change in Feed Flow Rate With Q-WT (top: $k_p = 1.5$ $k_i = .23$ bot: $k_p = .90$ $k_i = .47$)

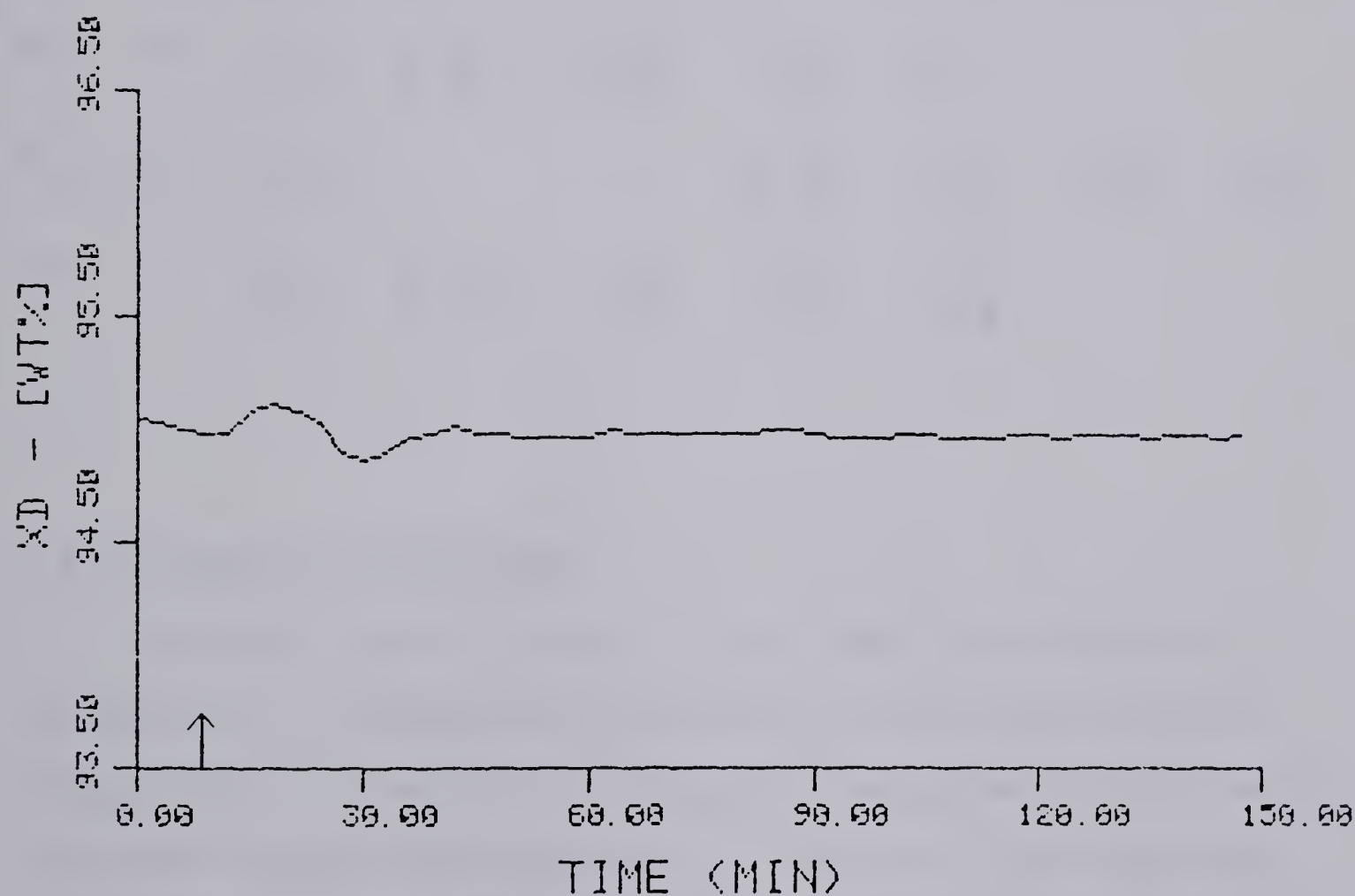
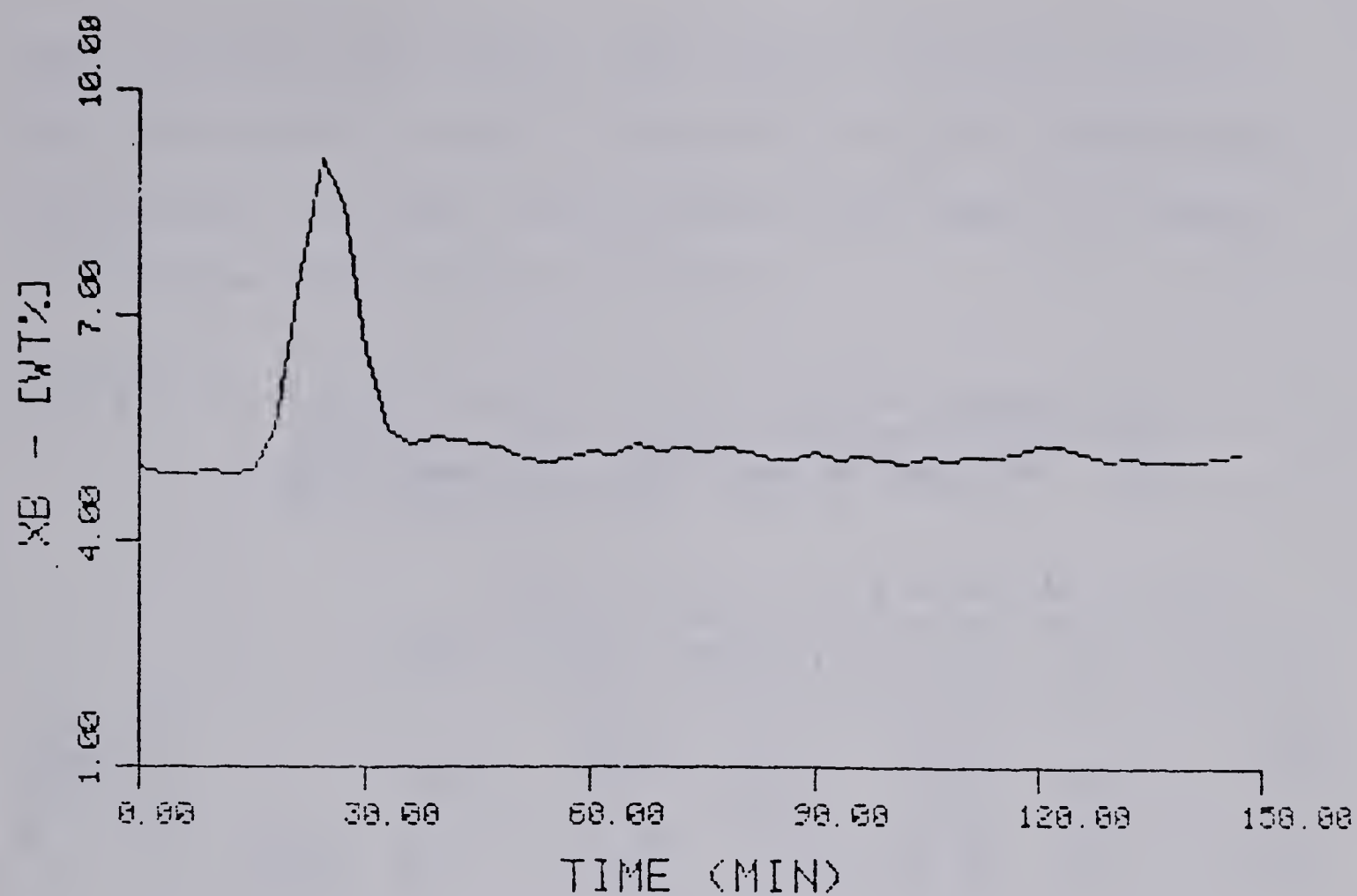


Figure 5.40 MV-ST Control of Terminal Compositions for a Step Increase in Feed Flow Rate to its Normal Steady State Value With Q-WT (top: $k_p = 1.5$ $k_i = .23$ bot: $k_p = .90$ $k_i = .47$)

about one-half the values that result using PID control. The improvement is due to the addition of the feedforward terms which link the two controllers and serve to reduce interaction hence improving control.

Table 5.11 Comparison of Top and Bottom Composition Absolute Error Values for Multiloop PID, Multiloop STC (With and Without FF), and Multivariable STC

Type of Control		T y p e O f D i s t u r b a n c e					
		-25% to Feed		Return to S.S.		+25% to Feed	
		Figure	SAE (wt%)	Figure	SAE (wt%)	Figure	SAE (wt%)
ML - PID	top	5.11	2.13	5.12	2.30	5.9	2.75
	bot	5.11	21.69	5.12	22.96	5.9	27.26
ML - STC	top	5.35	2.32	5.36	2.17		
	bot	5.35	23.75	5.36	19.71		
ML - STC + FF	top			5.37	3.74	5.38	2.27
	bot			5.37	14.59	5.38	8.97
MV - STC	top	5.39	1.38	5.40	1.07		
	bot	5.39	27.39	5.40	17.28		

5.8 DISCUSSION OF RESULTS

From the results given, it has been shown that the feedback self-tuning controllers can control the terminal compositions of the distillation column as well as or better than well tuned PID compensators. With the multivariable STC, the top composition controller is able to react immediately to steam changes hence improving the top composition control compared to the multiloop PID case.

The ability of the feedback STC to provide a control performance that is comparable to a PID compensator is remarkable considering the type of disturbance used to compare the controllers. The use of step changes in feed flow rate as disturbances is a very severe test. In the derivation of the STC, the disturbances have been assumed to have zero mean. When a step change in the feed flow rate is introduced, a new steady state control output is required, as calculated from the control law. Therefore, the feed flow step changes are not zero mean disturbances. To further appreciate this situation, consider what happens to the STC when the feed rate is increased. As the additional flow of material moves through the column, the bottom methanol composition starts to increase and the STC compares the composition with the predicted value and finds there is an error. The controller will then increase the steam flow rate and in addition change the parameters so that they will give a prediction consistent with the higher recorded value. As more of the increased feed reaches the reboiler, the composition becomes even higher, the steam flow rate is again increased and the parameters further changed to give the correct output. In effect, there is a transition period when the parameters change and the predicted output no longer matches the actual output. Figures 5.41 and 5.42 show how the bottom STC parameters change when a -25% step change in feed flow rate occurs (run shown in Fig 5.27). It is seen that there is a significant change in the parameter

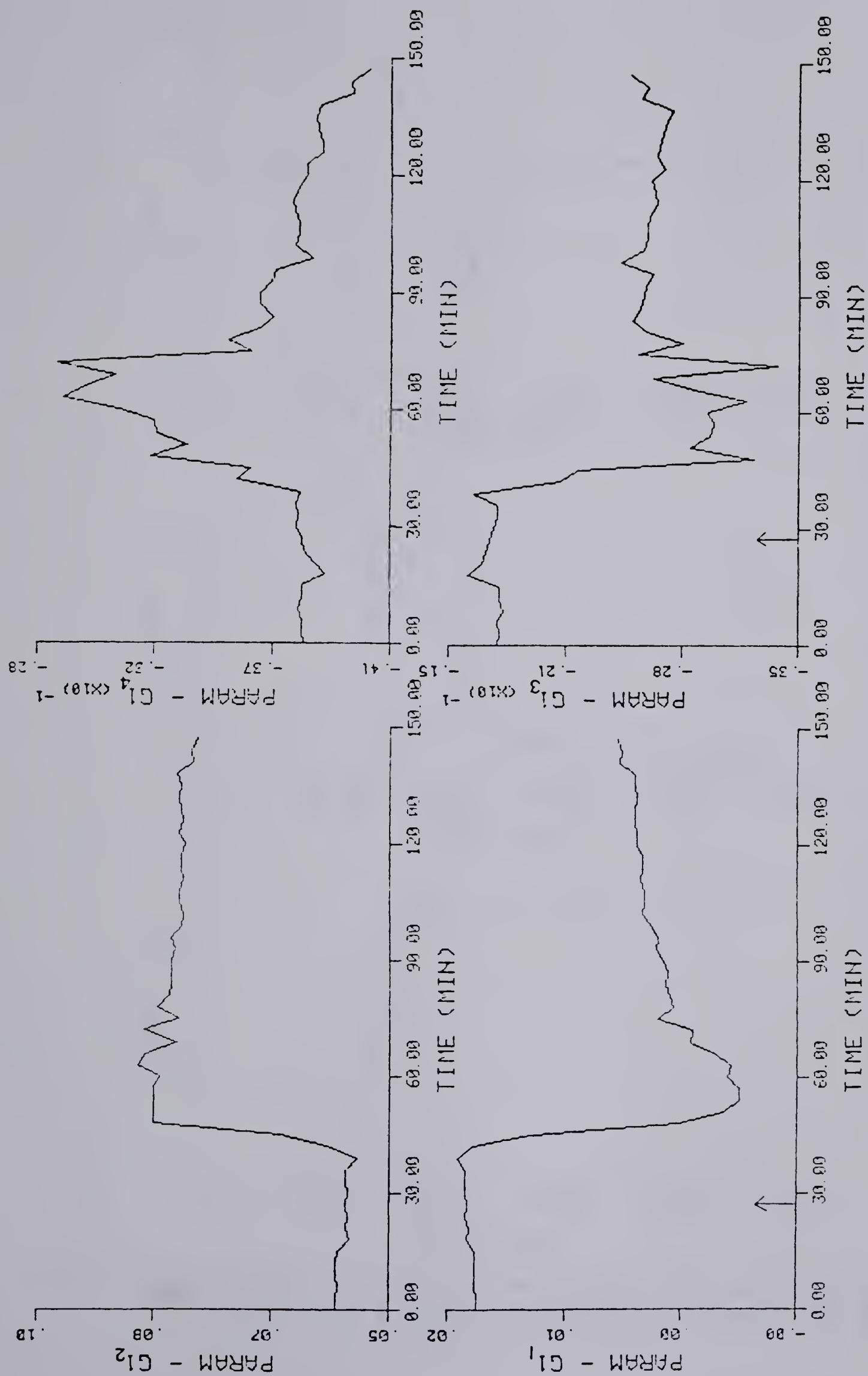


Figure 5.41 Variation of G1 Parameters in the Transition Period of a -25% Step Change in Feed Flow Rate for the Bottom SIC with Q-WT

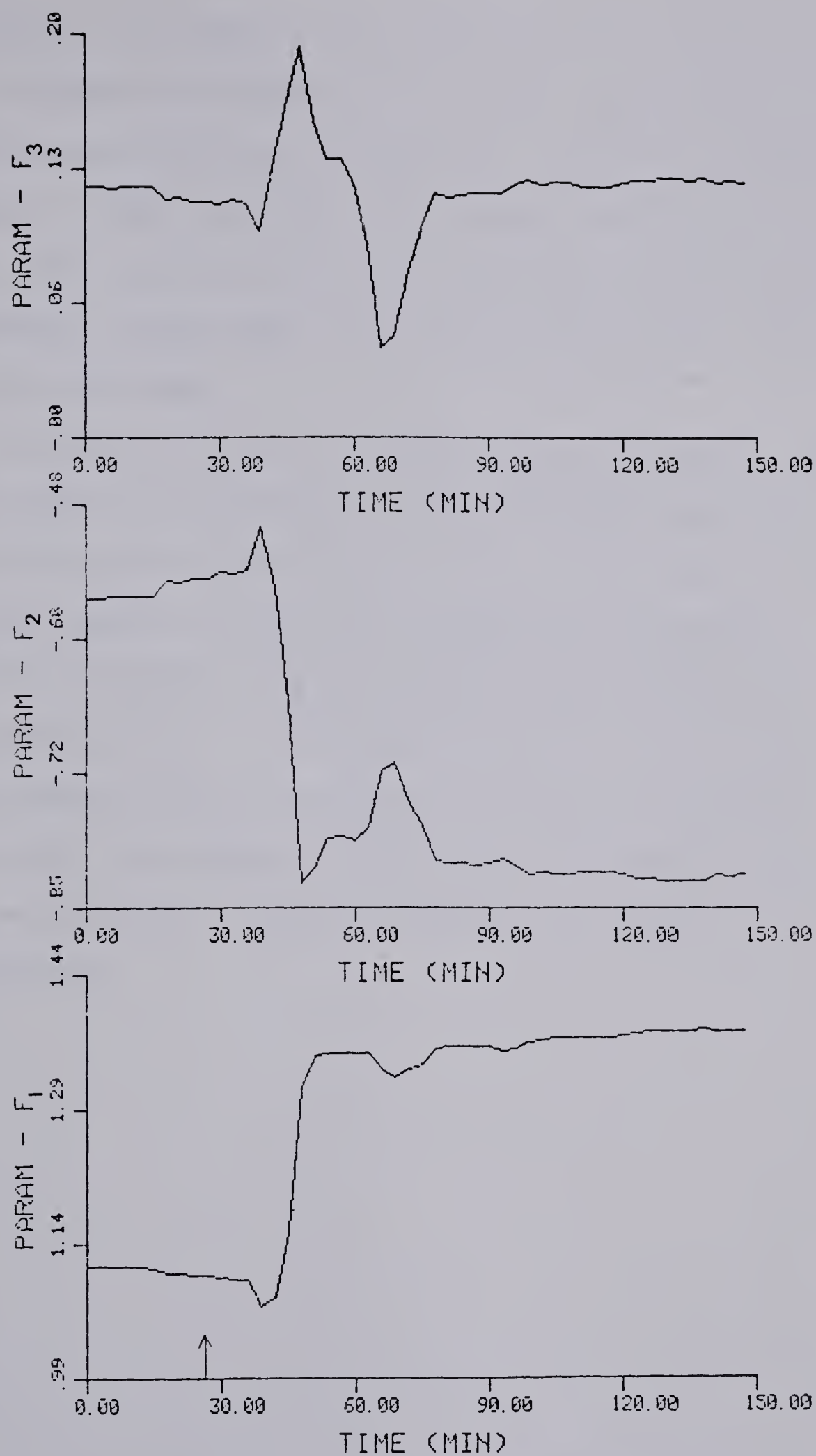


Figure 5.42 Variation of F Parameters in the Transition Period of a -25% Step Change in Feed Flow Rate for the Bottom STC with Q-WT

estimates. It is due to the robustness of the STC that using only feedback information, it is able to match the control performance of the well tuned PID. To properly handle step changes in feed flow rate, the feedforward option should be used. The results of the self-tuning controller with feedforward added does indeed show considerable improvement over the PID case.

It should also be noted that the PID control behavior shown has been obtained after considerable time was spent fine tuning the compensator. On the other hand, the amount of time needed to tune the Q-weighting was minimal. The original estimates obtained from Halman's technique [75] gave good control. With more tuning of the Q-weighting, it should be possible to improve the control performance. Additional improvement in the controller performance should also be realized if derivative action is added into the Q-weighting.

6. CONCLUSIONS AND RECOMENDATIONS

This work was concerned primarily with the experimental evaluation of the self-tuning controller. However, before the evaluation could be carried out, the pairing of manipulated and controlled variables had to be decided. The main concern was whether top composition should be controlled by manipulating reflux flow (energy balance control) or distillate flow (material balance control). It was found that compared to energy balance control, the material balance control strategy is not as sensitive to heat disturbance provided the external reflux ratio is large. However just the reverse case is true if the external reflux ratio is small. Analysis of the sensitivity of the vapor-to-liquid ratio to changes in the vapor rates for the two strategies provides valuable information on which of the two control schemes should provide better control performance for heat disturbances entering the column. For the pilot plant distillation column used in this work, analysis of the vapor-to-liquid ratio indicated that energy balance control was superior and this was substantiated by the results from experimental tests.

To evaluate the STC, a comparison was made between the performance of the STC and a well tuned PID compensator in controlling the terminal compositions of the pilot plant column when it was subjected to step changes in the feed rate. The evaluation was carried out for: top composition control, bottom composition control and dual composition

control. In all cases, it was found that the STC is superior to the PID compensator. Specifically, it has been shown that:

- (1) The STC acting only in the feedback mode using initial Q-weighting estimates from Halman's [75] technique performed as well as a PID compensator that had been very well tuned at a considerable expenditure of time.
- (2) Addition of feedforward control action on the bottom composition loop significantly improved the controller performance.
- (3) Use of the multivariable STC allowed the top composition controller to react immediately to changes in heat input hence improve top composition control performance.

In addition, the STC was found to have the following properties:

- (1) Starting with zero estimates, the STC can, for a fast system (top composition loop), converge to reasonable estimates after only 30 sampling intervals. It gives superior control compared to PID control results when the STC is used with these estimates.
- (2) Fixing the h_0 parameter value to -1 does not create any problems with parameter convergence.
- (3) The addition of Q-weighting reduces the variation in the manipulated variables and as a result the control of the output variable is improved considerably.
- (4) The controller parameters are able to readjust to some

degree for changes in the Q-weighting so that the output response is not as sensitive to changes in the Q-weighting constants as in the case of the PID compensator. Nevertheless, the Q-weighting can still be fine tuned the same way as a normal PI compensator is tuned to obtain even improved control responses. The relationship between K_p and K_i and how they affect the output response remains the same.

- (5) If the identification phase of the STC is stopped, an unknown disturbance will cause an offset to occur. The magnitude of the offset depends upon the sum of the F and G parameters.
- (6) To an extent, a large sampling time has the same effect as adding more Q-weighting; i.e., it smooths out the variation in the manipulated variable. However, if the STC is acting only in the feedback mode, the controller response will suffer because it reacts more slowly to unknown disturbances due to the larger sampling time.

In summary, the experimental evaluation of the self-tuning controller as modified by Morris [24] has shown that the controller is superior to the PID compensator. If tuning is necessary, the controller can be fine tuned the same way a PID regulator is tuned. The controller is extremely robust as it is able to handle non-zero mean disturbances even though the control law was derived based on the assumption that the disturbance is a white noise sequence.

From the results of this work, the following recommendations are given for possible future studies:

- (1) With the present column operating conditions, energy balance control has been shown to be superior. It is recommended that test be performed with the operating conditions altered so that the external reflux ratio is higher. This would indicate whether material balance control will indeed be less sensitive to heat disturbances under the new operating conditions.
- (2) While performing the experimental work for this study, observations on the multipoint temperature chart recorder indicated that the liquid temperature on tray 1 gave a good indication of changes in bottom composition. It is suggested that bottom composition be controlled by maintaining tray 1 temperature constant using the self-tuning controller. For a binary system, providing the pressure is constant, fixing the temperature implies that the composition will remain constant. To account for any variation in pressure, the result of the GC analysis can be used to reset the tray 1 temperature set point.
- (3) The self-tuning controller is designed primarily to handle stochastic disturbances while the evaluation has been carried out using deterministic disturbances. A further evaluation should be made on how the performance of the STC compares with the PID compensator when random disturbances are used; e.g.,

add noise to the composition measurements.

- (4) When an unknown deterministic disturbance is encountered, the STC forces the parameters to change rapidly to account for the disturbance. An improved response might be achieved by freezing the parameters once reasonable estimates have been obtained. To prevent the offset that occurs when identification is stopped, an integrator can be added to the controller. The control action would be the normal STC algorithm output plus an added action based on the error from set point.
- (5) In the present implementation of the STC algorithm, the feedforward control action option only allows the use of lead terms. As lag action is normally required in feedforward compensation, it should be added into the algorithm.
- (6) This study considered the performance of the self-tuning controller only for load disturbances. Additional testing should be performed to study the performance of the controller for set point changes.
- (7) The multivariable approach given by Morris [24] which is used in this study differs from that proposed by Keviczky [19] and Borrison [19] in that the identification of the system is performed on a loop by loop basis. Both Keviczky and Borrison suggested identifying the multivariable model directly. A direct comparison by simulation or experimentally of these

different multivariable self-tuning schemes should be undertaken.

- (8) Rather than starting with zero initial estimates, it is suggested that the STC be allowed to identify parameter estimates while the process is operating under PID control. These estimates should provide a smooth transition during the initial switch to the self-tuning controller.
- (9) The problem of the covariance matrix bursting due to inadequate excitation should be investigated. The different methods that have been proposed to overcome the problem should be tested; eg. use of recursive learning identification as proposed by Morris [24].

NOMENCLATURE

a. Abbreviations:

DACS - Data Acquisition Control and Simulation
DDC - Direct Digital Control
FF - FeedForward
GC - Gas Chromatograph
HP - Hewlett Packard
ML - Multiloop
MV - Multivariable
PID - Proportional-Integral-Derivative
Q-WT - Q-weighting
SAE - Sum of Absolute Error
SISO - Single Input Single Output
STR - Self-Tuning Regulator
STC - Self-Tuning Controller
S.S. - Steady State

b. Variables:

D - Distillate flow rate
FE - Feed flow rate
J - Cost function for SISO case
J - Cost function for multivariable case
I - Identity matrix
k - System time delay in integer sample intervals.
 k^A - Time delay due to other output for a multivariable system

- k^B - Time delay due to other control input for a multivariable system
- k^L - Time delay associated with load disturbance
- k_p - Proportional constant in PID compensator
- k_i - Integral gain in PID compensator
- k_d - Derivative gain in PID compensator
- \underline{K} - Kalman gain vector used in parameter estimation algorithm
- K1 - Use in STC program to designate time delay of system
- K2 - Use in STC program to designate time delay due to other control input for a multivariable system:
 $k_{ij}^L - k_{ii}^B$
- K3 - Use in STC program to designate time delay associated with load disturbance: $k^L - k$
- L - Liquid rate inside the column
- M - Number of loops in multivariable system
- n - System order
- ND - No. of D parameters in controller
- NG1 - No. of G1 parameters in controller
- NG2 - No. of G2 parameters in controller
- NF - No. of F parameters in controller
- NH - No. of H parameters in controller
- \underline{P}^C - Covariance Matrix
- RE - Reflux flow rate
- ST - Steam flow rate
- t - Time measured in integral sample intervals
- u - Control input
- V - Vapor rate inside the column
- v - Load disturbance

- \bar{v} - Modified load disturbance defined as $z^{k-k^L} v$
- w - Set point
- \bar{w} - Modified set point defined as Hw
- \underline{x} - Vector containing measured data
- x_B - Bottom Composition in WT% methanol
- x_D - Overhead Composition in WT% methanol
- y - System output
- \bar{y} - Modified system output defined as y/P_d
- z^{-1} - Backshift operator; $z^{-k} u_t = u(t-k)$

c. Greek:

- ϵ - Weighted output prediction error
- ξ - Uncorrelated zero-mean random input
- \emptyset - Scalar output function
- $\underline{\theta}$ - Vector of controller parameters: F, G, H, D
- ρ - Forgetting factor
- σ - Variance

d. Polynomials in z^{-1} :

- A - Order N corresponding to system output, $a_0 = 1$
- B - Order N corresponding to control input, $b_0 = 0$
- C - Order N corresponding to random disturbance, $c_0 = 1$
- D - Associated with load in final control law
- E - Use to separate past and future disturbance
- F - Use to separate past and future disturbance
- G - Associated with control input in the final control law

- H - Associated with set point in the final control law
- L - Order N corresponding to deterministic load disturbance input, $l_0 = 1$
- P - Weighting function on system output
- Q - Weighting function on system input
- R - Weighting function on set point

e. Superscript:

- ' - Indicates polynomials used for derivation of STC for the case without weighted output
- ˆ - Estimates of the parameters
- * - Predicted value
- T - Transpose of Matrix

f. Subscript:

- ij - For the multivariable case, denotes the jth component of the ith loop
- N - Numerator of a z transfer function
- D - Denominator of z transfer function
- t - Value at present time
- t+k - Value at k samples from present time

9. Symbols:

- [] - Use to designate the variables when extending the results to the multivariable system. When used with polynomials in z, the result is a polynomial matrix in z. When used with scalars, the results in a vector.
- - Designates a vector
- = - Designates a matrix

REFERENCES

1. Rademaker, O., Rijnsdorp, J.E., Maarleveld, A., "Dynamics and Control of Continuous Distillation Units", Elsevier Scientific Publishing Company, New York (1975).
2. Svrcek, W.Y., "Binary Distillation Column Dynamics", Ph.D. Thesis, University of Alberta, Edmonton, Alberta, 1967.
3. Chanh, B.M., "Control of a Binary Distillation Column: Effect of Sensor Location", M.Sc. Thesis, University of Alberta, Edmonton, Alberta, 1972.
4. McGinnis, R.G., "Multivariable Control of a Binary Distillation Column", M.Sc. Thesis, University of Alberta, Edmonton, Alberta, 1972.
5. Berry, M.W., "Terminal Composition Control of a Binary Distillation Column", M.Sc. Thesis, University of Alberta, Edmonton, Alberta, 1973.
6. Pacey, W.C., "Control of a Binary Distillation Column: An Experimental Evaluation of Feedforward and Combined Feedforward-Feedback Control Scheme", M.Sc. Thesis, University of Alberta, Edmonton, Alberta, 1973.
7. Liesch, D. W., "Decoupled Feedforward-Feedback Control of a Binary Distillation Column", M.Sc. Thesis, University of Alberta, Edmonton, Alberta, 1974.
8. Meyer, C.B.G., "Experimental Evaluation of Predictor Control Schemes for Distillation Column Control", M.Sc. Thesis, University of Alberta, Edmonton, Alberta, 1977.

9. Bilec, R.J., "Modelling and Dual Control of a Binary Distillation Column", M.Sc. Thesis, University of Alberta, Edmonton, Alberta, 1980.
10. Åström, K.J and Wittenmark, B., "On Self-Tuning Regulators", Automatica, 9, pp. 185-199, 1973.
11. Åström, K.J., Borisson, U., Ljung, L. and Wittenmark, B., "Theory and Applications of Self-Tuning Regulators", Automatica, 13, pp.457-476, 1977.
12. Peterka, V., "Adaptive Digital Regulation of Noisy Systems", Preprint of 2nd Prague IFAC Symp. on Identification and Process Parameter Estimation, Prague, paper no. 6.2, June, 1967.
13. Clarke, D.W., Cope, S.N. and Gawthrop, P.J., "Feasibility Study of the Application of Microprocessors to Self-Tuning Controllers", OUEL Research Report 1137/75, Department of Engineering Science, University of Oxford, Oxford (1975).
14. Clarke, D.W. and Gawthrop, P.J., "Self-Tuning Controllers", Proc. IEEE, 122, no. 9, pp. 929-934, 1975.
15. Clarke, D.W. and Gawthrop, P.J., "Simulation of a Generalized Self-Tuning Regulator", Electronic Letters, 11, pp. 41-42, Jan. 1975.
16. Clarke, D.W. and Gawthrop, P.J., "Generalization of the Self-Tuning Regulator", Electronic Letters, 11, pp. 40-41, Jan. 1975.
17. Gawthrop, P.J., "Some Interpretations of the

- Self-Tuning Controller", Proc. IEEE, 124, no. 10, pp. 889-894, 1977.
18. Clarke, D.W. and Gawthrop, P.J., "Self-Tuning Controllers", Proc. IEEE, 126, no. 6, June 1979.
 19. Borisson, U., "Self-Tuning Regulators for a Class of Multivariable Systems", Automatica, 15, pp. 209-215, 1979.
 20. Keviczky, L. and Hetthessy, J., "Self-Tuning Minimum Variance Control of MIMO Discrete Time Systems", Auto. Control, 5(1), Jan. 1977.
 21. Keviczky, L., Hetthessy, J., Hilger, M. and Kolostori, J., "Self-Tuning Adaptive Control of a Cement Raw Material Blending", Automatica, 14, pp. 525-532, 1978.
 22. Morris, A.J., Fenton, T.P., and Nazer, Y., "Application of Self-Tuning Regulators to the Control of Chemical Processes", IFAC Congress on Digital Computer Applications to Process Control, Holland, pp. 447-455, 1977, (V.N. Lenke, Editor).
 23. Morris, A.J. and Nazer, Y., "Distillation Column Control Using Self-Tuning Techniques", International Congress on the Contribution of Computers to the Development of Chemical Engineering and Industrial Chemistry, Paris, 1978.
 24. Morris, A.J. and Nazer, Y., "Self-Tuning Process Controllers for Single and Multivariable Systems", Submitted to Automatica, 1979.

25. Kalman, R.E., "Design of a Self-Optimising Control System", Trans. ASME, pp. 468-478, 1958.
26. Åström, K.J. and Bohlin, T., "Numerical Identification of Linear Dynamic Systems from Normal Operating Data", Proc. 2nd IFAC Symp. on the Theory of Self-Adaptive Control Systems, Teddington, pp. 96-111, Sept. 1965.
27. Åström, K.J., "On the Achievable Accuracy in Identification Problems", Preprints of 2nd IFAC Symp. on Identification in Automatic Control Systems, Prague, paper no. 1.8, June 1967.
28. Åström, K.J. and Wittenmark, B., "Problems of Identification and Control", Journal of Mathematical Analysis and Application, 34, pp. 90-13, 1971.
29. Wittenmark, B., "A Self-Tuning Regulator", Research Report 7311, Division of Automatic Control, Lund Institute of Technology, April, 1973.
30. Chang, F.L.Y, "An Application of Self-Tuning Regulators", M.Sc. Thesis, Dept. of Chemical Engineering, University of Alberta, Edmonton, Alberta, 1975.
31. Cegrell, T. and Hedqvist, T., "A Self-Adjusting Regulator", ASEA LME Automation, Internal report, 1973.
32. Åström, K.J., "Introduction to Stochastic Control Theory", Academic Press, New York (1970).
33. Åström, K.J. and Wittenmark, B., "Analysis of Self-Tuning Regulator for Nonminimum Phase Systems", Research Report 7418(c), Division of Automatic Control,

Lund Institute of Technology, August 1974.

34. Turtle, D.P. and Phillipson, P.H., "Simultaneous Identification and Control", *Automatica*, 7, pp. 445-453, 1971.
35. Gustavsson, I., Ljung, L. and Söderström, T., "Identification of Linear Multivariable Process Dynamics Using Closed Loop Experiments", Research Report 7401, Division of Automatic Control, Lund Institute of Technology, Jan. 1974.
36. Peterka, V. and Astrom, K.J., "Control of Multivariable System With Unknown but Constant Parameters", 3rd IFAC Symp. on Identification and System Parameter Estimation, Hague, pp.535-554, June, 1973.
37. De Keyser, R. and Van Gauwenberghe, A., "Self-Adjusting Control of Multivariate Nonstationary Process", Internal Report, Automatic Control Laboratory, University of Ghent, Belgium, 1978.
38. Sage, A.P. and Melsa, J.L., "System Identification", Academic Press, New York (1971).
39. Panuska, V., "An Adaptive Recursive Least Squares Identification Algorithm", Proc. 8th IEEE Symp. on Adaptive Processes, 1969.
40. Wittenmark, B. and Borisson, U., "An Industrial Application of a Self-Tuning Regulator", Research Report 7310, Division of Automatic Control, Lund Institute of Technology, April 1973.
41. Cegrell, T. and Hedqvist, T., "Successful Adaptive

Control of Paper Machines", Automatica, 11, pp. 53-59, 1975.

42. Borisson, U. and Syding, R., "Self-Tuning Control of an Ore-Crusher", Proceedings of the IFAC Symp. on Stochastic Control, Budapest, pp. 491-497, 1974.
43. Dummont, G.A. and Belanger, P.R., "Control of Titanium Dioxide Kilns", IEEE Trans. on Automatic Control, AC-23, no. 4, pp. 521-531, 1978.
44. Dummont, G.A. and Belanger, P.R., "Self-Tuning Control of A Titanium Dioxide Kiln", IEEE Trans. on Automatic Control, AC-23, no. 4, pp. 532-538, 1978.
45. Westerlund, T., Toivonen, H. and Nyman, K.E., "Stochastic and Self-Tuning Control of a Continuous Cement Raw Material Mixing System", Report 78-3, The Institution for Automatic Control, Finland, August 1978.
46. Sastry, V.A., Seborg, D.E. and Wood, R.K., "A Self-Tuning Regulator Applied to a Binary Distillation Column", Automatica, 13, pp. 417-424, 1977.
47. Harris, T.J., MacGregor, J.F. and Wright, J.D., "An Application of Self-Tuning Regulators To Catalytic Reactor Control", JACC, 2, 1978.
48. Jensen L. and Hänsel, R., "Computer Control of an Enthalpy Exchanger", Research Report 7417, Division of Automatic Control, Lund Institute of Technology, 1974.
49. Cegrell, T. and Hedqvist T., "A New Approach to

- Continuous Digester Control", IFAC Symp. on Digital Computer Applications to Process Control, Zurich, 1974.
50. Fortescue, T.R., Kerschenbaum, L.S. and Ydstie, B.E., "Implementation of Self-Tuning Regulators with Variable Forgetting Factor", Submitted for Publication, 1979.
 51. Källström, C.G. and Astrom, K.J., "Adaptive Autopilots for Large Tankers", Proc. of the IFAC World Congress, Helsinki, 1978.
 52. Ljung, L., "Convergence Recursive Stochastic Algorithms", Proc. of the IFAC Symp. on Stochastic Control, Budapest, pp.215-223, 1974.
 53. Ljung, L., "Convergence of Recursive Stochastic Algorithms", Research Report 7403, Division of Automatic Control, Lund Institute of Technology, 1974.
 54. Ljung, L., "Consistency of the Least-Squares Identification Method", IEEE trans., AC-21, pp. 779-781, 1976.
 55. Ljung, L., "Analysis of Recursive Stochastic Algorithms", IEEE trans., AC-22, pp. 551-575, 1977.
 56. Ljung, L., "On Positive Real Transfer Functions and the Convergence of Some Recursive Scheme", IEEE trans., AC-22, pp.539-550, 1970.
 57. Landau, I.D., "Unbiased Recursive Identification Using Model-Reference Adaptive Techniques: Theory and Applications", IEEE Trans., AC-21, pp. 194-202, 1976.
 58. Landau, I.D., "An Addendum to Unbiased Recursive Identification Using Model-Reference Adaptive

- Techniques", IEEE Trans., AC-23, pp. 97-99, 1978.
59. Chung, K.L., "A Course in Probability Theory", 2nd Edition, Academic Press (1974).
 60. Doob, J.L., "Stochastic Processes", Wiley (1953).
 61. Gawthrop, P.J., "On the Stability and Convergence of Self-Tuning Controllers", Proc. of the IMA Conference on Analysis and Optimization of Stochastic Systems, 1978.
 62. MacGregor, J.F. and Tidwell, P.F., "Discrete Stochastic Control With Input Constraints", Proc. IEE, 124, pg. 732, 1977.
 63. Meyer, C., Seborg, D.E. and Wood, R.K., "Experimental Application of Time Delay Compensation Techniques to Distillation Column Control", IFAC Congress on Digital Computer Applications to Process Control, Holland, 1977.
 64. Smith, O.J.M., "A Controller to Overcome Dead Time", ISA Journal, 6, no. 2, pp.28-33, 1959.
 65. Hastings-James, R. and Sage, M.W., "Recursive Generalized Least Square Procedure for Online Identification of Process Parameters", Proc. IEEE, 116, no. 12, DEC. 1969.
 66. Kan, H., "Binary Distillation Column Control: Evaluation of Digital Control Algorithms", M. Sc. Thesis, University of Alberta, Edmonton, Alberta, 1980.
 67. Shinskey, F.G., "Distillation Control for Productivity and Energy Conservation", McGraw-Hill, New York, 1977.

68. Bristol, E.H., "On a New Measure of Interaction for Multivariable Process Control", IEEE Trans. Auto. Control, AC-10, 133, 1965.
69. Juantoreno and Romeo, R.T., "An Application of the Relative Gain Matrix to a Distillation Column", Proc. 12th Annual ISA Chem. and Petrol. Instrum. Symp., Houston, Texas, 1971.
70. McClune, L.C., Gallier, P.W., "Digital Simulation: A tool for the Analysis and Design of Distillation Controls", ISA Trans., 12, 193, 1973.
71. Nisenfeld, A.E., "Reflux or Distillate: Which to Control", Chem. Eng., 76, 169, 1969.
72. Van Kampen, J.A., "Automatic Control by Chromatographs of the Product Quality of a Distillation Column", 3rd IFAC Congress, London, 1966.,
73. McNeill, G.A. and Sacks, J.D., "High Performance Column Control", Chem. Eng. Prog., March 1969.
74. Rosenbrock, H.H., "The Control of Distillation Columns", Trans. Instn.. Chem. Engrs., 40, 35, 1962.
75. Verbruggen, J.A., Peperstraete, J.A. and Debruyne, H.P., "Digital Controllers and Digital Control Algorithms", Journal A, 16, 2, pp.53-67, 1975.

APPENDIX A

DETAIL SCHEMATIC OF PILOT SCALE DISTILLATION COLUMN

LEGEND:



- indicates local analog equipment

TYPE - gives function

TYPE CODES

T - transmitter

R - recorder

C - controller

I - indicator

F - flow

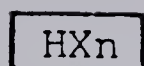
A - concentration (analyzer)

T - temperature

P - pressure

DP - differential pressure

n - identification number



- heater exchanger

n - identification number



- gas chromatograph



- thermocouple



- indicates utilities

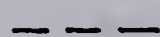
UT - ST - 60 psig steam

- CW - cooling water



- solenoid valve

n - identification number



- control lines



- process lines

APPENDIX B

SAMPLE CALCULATION TO OBTAIN INITIAL Q-WEIGHTING PARAMETERS

1. Using Halman's technique [75], the k_p and k_i are related to the process dynamics according to:

$$a = 1 + 0.5 * T_s / T_d \quad (B.4)$$

$$k = 2\tau / 3K_G a T_s \quad (B.5)$$

$$a_0 = 1 + T_s / 2\tau \quad (B.6)$$

$$a_1 = -(1 - T_s / 2\tau) \quad (B.7)$$

$$k_p = -a_0 * k \quad (B.8)$$

$$k_i = (a_0 + a_1) * k \quad (B.9)$$

Where:

k_p = proportional gain constant in Q-wt
 k_i = integral gain constant in Q-wt
 K_G = Process gain
 T_d = Process time delay
 T_s = Sample time
 τ = Process time constant

2. From results given by Kan [66], the bottom composition response for a 7 1/2% decrease in steam can be represented by a first order plus time delay model with constants:

$K_G = 1.965 \text{ wt\%/(g/s)}$
 $\tau = 1000 \text{ second}$
 $T_s = 180 \text{ second}$
 $T_d = 316 \text{ second}$

3. From the discussion given in Chapter 2, it has been shown that the affect of the process time delay should be removed. So, the time delay is set equal to the sample time. Using the values given in section 2, the constants in section 1 become:

$a = 1.5$
 $k = 1.259$
 $a_0 = 1.09$
 $a_1 = -.91$

The controller constants are then:

$$\begin{aligned}k_p &= 1.145 \\k_i &= 0.226\end{aligned}$$

APPENDIX C

PROGRAMS LISTING

PROGRAM	D E S C R I P T I O N	PAGE
STC	Implements the STC Algorithm on the Column	151
HSET	Sets Original Controller Parameters and Option	178
ERSUM	Computes Absolute Error Value of Data	186
HLOT	Plots Data Stored in Files	188
GCLNK	Obtains GC Report from GC Computer	196
GCSCH	Process GC Report and Initiate GC Cycle	198
GCSWT	Reset Digital Switch That Initiate GC Cycle	201
BDC99	Monitors Column Operation	202


```

0001 C
0002 FTN4
0003 PROGRAM STC ( ,94) ,ST CONTROL MOD. BY HYL NOV. 1, 1979
0004 COMMON ICOM(1)
0005 DIMENSION IDCB2(144) , IDCB1(144) , IFILE(3) , IFILW(3)
0006 DIMENSION ICX(5) , IDX(6) , ERR1(2) , ERR2(2) , IPRAH(2)
0007 DIMENSION PERERR(2),YSTAR(2),USEND(2)
0008 DIMENSION YERR(2),DPR(2),BUF(56),ACLOSS(2)
0009 DIMENSION NSELF(2),N(2,6),ITYID(2),SAVE(2,11,30)
0010 DIMENSION WN(2),DN(2),YN(2),WO(2),WOLD(2),PHI(2)
0011 DIMENSION K1(2),K2(2),K3(2),UN(2),CA(2),CB(2),CO(2)
0012 DIMENSION NQ(2),QD(2,4),QN(2,4)
0013 DIMENSION NP(2),PN(2,4),PD(2,4)
0014 DIMENSION ITYW(2),IHW(2),HW(2),LW(2),BW(2)
0015 DIMENSION ITYD(2),IMD(2),HD(2),LD(2),BD(2)
0016 DIMENSION ITYCT(2),UNT(2),UBL(2),UTL(2),IRAMP(2),ACT(2)
0017 DIMENSION USS(2) , UPID(2)
0018 DIMENSION UPROP(2),UINT(2),UDINT(2),USINT(2),UDER(2)
0019 DIMENSION AKP(2) ,AKI(2) ,AKD(2) ,AKW(2),AKDN(2)
0020 DIMENSION AKPIS(2),AKPD(2),TS(2),UDIS(2)
0021 DIMENSION AKPS(2),AKIS(2),AKDS(2),ER1(2),ER2(2),SIAE(2)
0022 DIMENSION ADCC(2),ADCH(2),DACC(2),DACH(2),ITEST(2,8)
0023 DIMENSION KRS(16) , CHAR(5) , ITIME(5) , IOUTCH(2)
0024 DIMENSION X(2,30),PAR(2,30),P(2,210),G(2,30),ROW(2,3)
0025 DIMENSION IPIS(2),IP1(2),IP2(2),IFP(2,2),SADC(8)
0026 C
0027 C
0028 DATA NPARN/30/,NPMN/210/,NSCAN/5/,ISCAN/500/,
0029 * CA , CB , CO / 2*1. , 4*0.0/,
0030 * ICX/15,10,13,9,12/,
0031 * ILG,IFILE/110,2H&H,2HDA,2HTA/,
0032 * ITEST/16*0/,IOUTCH/7,8/,
0033 * CHAR/4HINC ,4HSTQ ,4HSTI ,4HPOS ,4HMAN /
0034 C
0035 C
0036 CALL RHPAR(IPRAH)
0037 IWIO=IPRAH(1)
0038 IRIO = IWIO
0039 C
0040 C-----
0041 C-
0042 C- OPEN UP FILE CONTAINING "ORIGINAL DATA".
0043 C-
0044 C
0045 ICR=136
0046 5 CALL OPEN(IDCB1,IERR,IFILE,0,77,ICR,144)
0047 C
0048 C-----
0049 C-
0050 C- INITIALIZING CONSTANTS.

```



```

0051 C-
0052 C
0053      M=0
0054      DO 25 I=1,2
0055 C
0056 C
0057      AKDN(I)=0.0
0058      ER1(I) =0.0
0059      IPIS(I)=1
0060      ER2(I) =0.0
0061      ERR1(I) =0.0
0062      ERR2(I) =0.0
0063      SIAE(I)=0.0
0064      UINT(I)=0.0
0065      USINT(I)=0.0
0066      YN(I) =0.0
0067      UN(I) =0.0
0068      UNT(I)=0.0
0069      YSTAR(I)=0.0
0070      WOLD(I)=0.0
0071      ACLOSS(I)=0.0
0072      AKNEW=0.0
0073      M=0
0074      IJK=0
0075      ITEST(I,8)=5
0076      ITEST(I,1)=1
0077 C
0078      DO 15 J=1,NPMN
0079      P(I,J)=0.0
0080      15 CONTINUE
0081 C
0082      DO 20 J=1,NPARN
0083      X(I,J)=0.0
0084      G(I,J)=0.0
0085      DO 20 K=1,11
0086      SAVE(I,K,J)=0.0
0087      20 CONTINUE
0088      25 CONTINUE
0089      WRITE(IWIO,800)
0090 C
0091 C-----
0092 C-
0093 C- ASK FOR NO. OF SYSTEM AND INTIAL FREQUENCY OF DUMPING
0094 C- INFORMATION TO DISK.
0095 C-
0096 C
0097      READ (IRIO,*) NSYSRD , IDMP , IDPR
0098      DO 90 IS=1,NSYSRD
0099 C
0100 C-----
0101 C-

```



```

0102 C- READING FROM SECTOR 1 [IS=1] AND 4 [IS=2].
0103 C-
0104 C
0105         CALL READF(IDCBI,IERR,BUF,80,ILG)
0106         TS(IS)=BUF(2)
0107         CALL READF(IDCBI,IERR,BUF,90,IDUM)
0108         ADCH(IS)=      BUF(40)
0109         ADCC(IS)=      BUF(41)
0110 C
0111 C
0112         DACH(IS)=      BUF(42)
0113         DACC(IS)=      BUF(43)
0114 C
0115 C-----
0116 C-
0117 C- READ FROM SECTOR 2 AND 5.
0118 C-
0119 C
0120         CALL READF(IDCBI,IERR,BUF,80,ILG)
0121 C
0122 C- NO. OF DELAYS FOR CONTROL, INTERACTIVE AND FEEDFORWARD
0123 C- TERM.
0124 C
0125         K1(IS) = IFIX(BUF(6))
0126         K2(IS) = IFIX(BUF(7))
0127         K3(IS) = IFIX(BUF(8))
0128 C
0129 C- OBTAIN INITIAL PARAMETER GUESS.
0130 C
0131         IFLG = 0
0132         NSELF(IS)=0
0133         IJ      =0
0134         DO 45 I=1,5
0135             IF(I .LT. 5) N(IS,I)=IFIX(BUF(I))
0136             NSELF(IS)=NSELF(IS)+N(IS,I)
0137             NLOOP      = N(IS,I)
0138             IF( NLOOP .EQ. 0) GO TO 45
0139             DO 40 J=1,NLOOP
0140                 IJ = IJ + 1
0141                 IK = I*10 + J - 1
0142                 IF(IK .GT. 40) IK = IK -40
0143                 PAR(IS,IJ)=BUF(IK)
0144                 IF(I.NE.4 .OR. J.GT.1) GOTO 40
0145                 N(IS,5) = IFIX(BUF(5))
0146                 CALL READF(IDCBI,IERR,BUF,90,IDUM)
0147                 IFLG = 1
0148         40          CONTINUE
0149         45          CONTINUE
0150 C
0151 C- TYPE OF IDENTIFICATION SCHEME.
0152 C

```



```

0153      IF(IFLG.NE.1) CALL READF(IDCBI,IERR,BUF,90,IDUM)
0154      ITYID(IS)=IFIX(BUF(20))
0155      ROW(IS,1)=      BUF(21)
0156      ROW(IS,2)=      BUF(22)
0157      ROW(IS,3)=      BUF(26)
0158      PINT      =      BUF(23)
0159      PINTI=PINT
0160 C
0161      IF(ITYID(IS).NE.2) GO TO 50
0162 C
0163 C- USING SQUARE ROOT ID METHOD.
0164 C
0165 C
0166 C
0167      PINT=SQRT(PINT)
0168      ROW(IS,1)=SQRT(ROW(IS,1))
0169      ROW(IS,2)=SQRT(ROW(IS,2))
0170      ROW(IS,3)=SQRT(ROW(IS,3))
0171      50      CONTINUE
0172      IFP(IS,1)=IFIX(BUF(24))
0173      IFP(IS,2)=IFIX(BUF(25))
0174      IF(ITYID(IS) .EQ. 3) GO TO 60
0175 C
0176 C- COVARIANCE MATRIX FOR ID TYPE 1,2, & 4.
0177 C
0178      IJ = 0
0179      IJP= 1
0180      NLOOP=NSELF(IS)
0181      DO 55 I = 1,NLOOP
0182      IJ = IJ + I
0183      IF(I.NE.IFP(IS,1).AND.I.NE.IFP(IS,2))
0184      *          P(IS,IJ) = PINT
0185      55      CONTINUE
0186      GO TO 70
0187 C
0188 C- COVARIANCE MATRIX INITIALIZATION FOR IDENTIFIATION
0189 C- METHOD 3.
0190 C
0191      60      IJ = 1
0192      NLOOP=NSELF(IS)
0193      IK = NLOOP
0194      DO 65 I=1,NLOOP
0195      IF(I.NE.IFP(IS,1).AND.I.NE.IFP(IS,2))
0196      *          P(IS,IJ) = PINT
0197      IJ = IK + 1
0198      IK = IJ + NLOOP - I - 1
0199      65      CONTINUE
0200      70      CONTINUE
0201 C
0202 C- INFORMATION ON SETPOINT.
0203 C

```



```

0204          ITYW(IS)=IFIX(BUF(30))
0205          HW(IS)  =      BUF(31)
0206          LW(IS)  =IFIX(BUF(32))
0207          BW(IS)  =      BUF(35)
0208          AKW(IS) =      BUF(35)
0209 C
0210 C-  INFORMATION ON LOAD.
0211 C
0212          ITYD(IS)=IFIX(BUF(40))
0213          HD(IS)  =      BUF(41)
0214          LD(IS)  =IFIX(BUF(42))
0215          BD(IS)  =      BUF(45)
0216 C
0217 C-----
0218 C-
0219 C-  READING FROM SECTOR 3 [IS=1] AND 6 [IS=2].
0220 C
0221 C
0222 C-
0223 C
0224          CALL READF(IDCBI,IERR,BUF,80,ILG)
0225          ITYCT(IS)=IFIX(BUF(1))
0226          IF(ITYCT(IS) .NE. 3) GO TO 80
0227              DO 75 II=1,6
0228                  N(IS,II)=0
0229                  PAR(IS,II)=0.0
0230          75      CONTINUE
0231          80      CONTINUE
0232 C
0233          UBL(IS)  =      BUF(2)
0234          UTL(IS)  =      BUF(3)
0235          IRAMP(IS)=IFIX(BUF(4))
0236          ACT(IS)  =      BUF(6)
0237          NQ(IS)   =IFIX(BUF(10))
0238          NP(IS)   =IFIX(BUF(30))
0239 C
0240          DO 85 I=1,4
0241              QD(IS,I)=BUF(I+10)
0242              QN(IS,I)=BUF(I+14)
0243              PD(IS,I)=BUF(I+30)
0244              PN(IS,I)=BUF(I+34)
0245          85      CONTINUE
0246 C
0247 C-  KP,KI,KD USED FOR PID AND Q-WEIGHTING.
0248 C
0249          CALL READF(IDCBI,IERR,BUF,90,IDUM)
0250          AKPS(IS)=BUF(21)
0251          AKIS(IS)=BUF(22)
0252          AKDS(IS)=BUF(23)
0253          AKP(IS)=BUF(24)
0254          AKI(IS)=BUF(25)

```



```

0255          AKD(IS)=BUF(26)
0256      90  CONTINUE
0257  C
0258  C-  READ IN PRESENT OUTPUT AND EQUATE USS TO PRESENT OUTPUT.
0259  C
0260          DO 88  IS = 1,NSYSRD
0261          IDUM = 13
0262          IF(IS .EQ. 2) IDUM = 9
0263          CALL AIRD(5,IDUM,IMEA,IERR)
0264          USS(IS) = 10. * SQRT(IMEA*ADCM(IS) + ADCC(IS))
0265          IF(ACT(IS) .LT. 0.)  USS(IS) = 100 - USS(IS)
0266          UN(IS) = USS(IS)
0267          UPID(IS) = USS(IS)
0268      88  CONTINUE
0269  C
0270  C-  CLOSE FILES CONTAINING INITIAL STARTING DATA.
0271  C
0272          CALL CLOSE(IDCBI,IERR)
0273  C
0274  C
0275  C
0276  C
0277  C-----
0278  C-
0279  C-  ASK FOR NAMES OF TWO FILES WHERE DATA WILL BE DUMPED.
0280  C-  OPEN FILE ACCORDINLY.
0281  C-
0282  C
0283      91  WRITE(IWIO,820)
0284          READ(IRIO,825)IFILW,ICR
0285          CALL OPEN(IDCBI,IERR,IFILW,0,77,ICR,144)
0286          IF(IERR.LT.0) CALL FMGER(IERR)
0287          WRITE(IWIO,820)
0288          READ(IRIO,825)IFILW,ICR
0289          CALL OPEN(IDCBI,IERR,IFILW,0,77,ICR,144)
0290          IF(IERR.LT.0) CALL FMGER(IERR)
0291          IDISK=IDMP
0292          ICNT=IDISK-1
0293          DO 94 I=1,56
0294      94  BUF(I)=0.0
0295          MULT=((IFIX(TS(1)))/NSCAN)
0296          MULT2 = IFIX(TS(2)/TS(1))
0297  C
0298  C*  IJK = MULT-1  WILL CAUSE EXECUTION OF CONTROL SECTION ON
0299  C*  FIRST SCAN.
0300  C
0301          IJK=MULT-1
0302          IJK2 = MULT2 -1
0303  C
0304  C-  ASK FOR DESCRIPTOR FOR EACH DATAFILE.  ONE LINE OF 80
0305  C-  ASCII CHARACTERS CAN BE USED.

```



```

0306 C
0307 DO 940 IS = 1, NSYSRD
0308 WRITE(IWIO,932) IS
0309 READ(IWIO,934) (BUF(I), I = 1,20)
0310 IF(IS .EQ. 2) GOTO 935
0311 CALL WRITF(IDCBI,IERR,BUF,40)
0312 GOTO 937
0313 935 CALL WRITF(IDCBI,IERR,BUF,40)
0314 937 CONTINUE
0315 940 CONTINUE
0316 C
0317 934 FORMAT(20A4)
0318 932 FORMAT(' ENTER DESCRIPTOR FOR',I3)
0319 C
0320 C* INITIALIZE THE START TIME
0321 C* ITM1 IS IN UNITS OF 10*MILLISECOND.
0322 C
0323 CALL EXEC(11,ITIME)
0324 ITM1=100*ITIME(2)+ITIME(1)
0325 C
0326 C
0327 C
0328 C*****
0329 C*
0330 C
0331 C
0332 C* START OF EACH SCAN
0333 C*
0334 C*****
0335 C
0336 C
0337 C-----
0338 C-
0339 C- READ STATUS OF REGISTER USING SUBROUTINE RSW. THE STATUS
0340 C- WILL BE STORED IN ARRAY KRS. KRS(1) WILL BE THE
0341 C- STATUS OF BIT 1 FROM FROM THE LEFT,
0342 C- I.E. BIT 15 AS INDICATED ON THE SWITCH.
0343 C- A VALUE OF 1 INDICATES THAT THE SWITCH IS UP, 0 IS FOR DOWN.
0344 C-
0345 C
0346 95 CONTINUE
0347 C
0348 CALL EXEC(11,ITIME)
0349 ITM=100*ITIME(2)+ITIME(1)
0350 CALL RSW(KRS)
0351 IF(KRS(15).EQ.1) WRITE(IWIO,1000) ITM,ITM1,IWAIT
0352 1000 FORMAT(' TIME IS = ',2I8,' IWAIT = ',I4)
0353 C
0354 C- DECIDE WHICH SYSTEM
0355 C
0356 IS=1

```



```

0357      IF(KRS(9).EQ.1) IS=2
0358      IF(KRS(1) .EQ. 1) GO TO 105
0359      IF(KRS(2) .EQ. 0) GO TO 105
0360      C
0361      C-----
0362      C-
0363      C- BIT 2 FROM LEFT IS ON, WANT TO PRINT OUT VALUES OF
0364      C- SPECIFIED CONSTANTS. (BIT2 FROM LEFT IS BIT 14 ON SWITCH
0365      C- REGISTER).
0366      C-
0367      C
0368      ASIG=+1.0
0369      IF(KRS(5).EQ.1) ASIG=-1.0
0370      AKNEW=AKNEW+ASIG*(FLOAT(KRS(6))+0.1*FLOAT(KRS(7))+
0371      &          0.01*FLOAT(KRS(8)) )
0372      IF(KRS(4).EQ. 1) AKNEW=0.0
0373      WRITE(IWIO,600)IS,AKNEW,AKP(IS),AKI(IS),AKD(IS)
0374      &      ,AKW(IS),AKDN(IS),AKPS(IS),AKIS(IS),AKDS(IS)
0375      GO TO 110
0376      C
0377      C-----
0378      C-
0379      C- CONVERT REGISTER READING TO TYPE OF ITEST AND THE NEW
0380      C- VALUE.
0381      C
0382      C
0383      C-
0384      C
0385      C
0386      C
0387      105  IF(KRS(1).EQ.0) GO TO 110
0388      IT = KRS(5)+KRS(4)*2+KRS(3)*4+1
0389      ITEST(IS,IT)=KRS(8)+KRS(7)*2+KRS(6)*4
0390      IF(ITEST(IS,1).EQ.7) WRITE(IWIO,815) IDX
0391      C
0392      C-----
0393      C-
0394      C- CHANGE VALUE OF CONSTANTS TO VALUE STORED IN AKNEW.
0395      C-
0396      C
0397      110  IF(KRS(10).EQ.1)WRITE(IWIO,610)((ITEST(JJ,J),J=1,8),
0398      *      JJ=1,2)
0399      IF(ITEST(IS,4) .EQ. 1) AKP(IS) =AKNEW + AKP(IS)
0400      IF(ITEST(IS,4) .EQ. 2) AKI(IS) =AKNEW + AKI(IS)
0401      IF(ITEST(IS,4) .EQ. 3) AKD(IS) =AKNEW + AKD(IS)
0402      IF(ITEST(IS,4) .EQ. 5) AKPS(IS)=AKNEW + AKPS(IS)
0403      IF(ITEST(IS,4) .EQ. 6) AKIS(IS)=AKNEW + AKIS(IS)
0404      IF(ITEST(IS,4) .EQ. 7) AKDS(IS)=AKNEW + AKDS(IS)
0405      C
0406      IF(ITEST(IS,5) .EQ. 1) AKW(IS)=AKNEW
0407      IF(ITEST(IS,5) .EQ. 2) AKDN(IS)=AKNEW

```



```

0408      IF(ITEST(IS,5) .EQ. 3) TPFIL=AKNEW
0409      IF(ITEST(IS,5) .EQ. 6) USS(IS)=AKNEW + USS(IS)
0410      IF(ITEST(IS,5) .EQ. 7) IRAMP(IS)=IFIX(AKNEW)+IRAMP(IS)
0411      IF(ITEST(IS,6) .EQ. 7) MULT=((IFIX(AKNEW))/NSCAN)
0412      C
0413      C- CHECK IF JUST WAITING, WANT TO STOP, READ INITIAL DATA
0414      C- FILE AGAIN.
0415      C
0416      IF(ITEST(1,8).EQ.5.OR.ITEST(2,8).EQ.5) GOTO 301
0417      IF(ITEST(IS,8).GE.6) GOTO 999
0418      C
0419      C
0420      C-----
0421      C-
0422      C- CHECK FOR NEW VALUE OF SETPOINT.
0423      C-
0424      C
0425      DO 157 IS=1,NSYSRD
0426          WO(IS)=BW(IS)
0427          IF(ITEST(IS,2) .EQ. 0) IMW(IS)=0
0428          IF(ITEST(IS,2) .EQ. 3) WO(IS) =AKW(IS)
0429          IF(ITEST(IS,2) .EQ. 6) WO(IS) =BW(IS)+11.67
0430          IF(ITEST(IS,2) .EQ. 7) WO(IS) =BW(IS)-11.67
0431      140      IF(ITEST(IS,4) .NE. 4) GO TO 156
0432      C
0433      C-----
0434      C-
0435      C- RESETTING THE Q-MATRIX
0436      C-
0437      C
0438          NQ(IS)=1
0439          DO 145 I=1,4
0440      C
0441      C
0442          QD(IS,I)=0.0
0443          QN(IS,I)=0.0
0444      145      CONTINUE
0445      C
0446      C- IF KRS(14) IS UP, THEN WANT TO REMOVE Q-FILTER.
0447      C
0448          IF(KRS(14).EQ.0) GO TO 147
0449          QD(IS,1)=1
0450          GO TO 155
0451      C
0452      C- RESETTING Q-MATRIX TO PID FILTER.
0453      C
0454      147      NQ(IS)=3
0455          QN(IS,1)=1.0
0456          QN(IS,2)=-1.0
0457          QD(IS,1) = AKP(IS) + AKI(IS) + AKD(IS)
0458          QD(IS,2) = -(AKP(IS) + 2*AKD(IS))

```



```

0459          QD(IS,3) = AKD(IS)
0460 C
0461 155      WRITE(IWIO,605)IS,AKP(IS),AKI(IS),AKD(IS)
0462      &          ,IS,NQ(IS),(QD(IS,J),J=1,4),(QN(IS,J),J=1,4)
0463 156      CONTINUE
0464 C
0465 C
0466          IF(ITEST(IS,5).EQ.0) GO TO 157
0467          IF(ITEST(IS,5).EQ.4) WRITE(IWIO,606)TPFIL
0468          IF(ITEST(IS,5).NE.5) GO TO 157
0469 C
0470 C-----
0471 C-
0472 C-  RESETTING THE P-WEIGHTING.
0473 C-
0474 C
0475          NP(IS)=1
0476          PD(IS,1)=1.0
0477          PN(IS,1)=1.0
0478          PN(IS,2)=0.0
0479          IF(TPFIL.LE.0.0) GO TO 157
0480          NP(IS)=2
0481          PN(IS,2)=-EXP(-1.0/TPFIL)
0482          PD(IS,1)=1.0-PN(IS,2)
0483          WRITE(IWIO,607)TPFIL,IS,NP(IS),(PN(IS,J),J=1,4),
0484      &          (PD(IS,J),J=1,4)
0485 157      WN(IS)=W0(IS)
0486 C
0487 C-----
0488 C-
0489 C-  RESETTING THE COVARIANCE MATRIX.
0490 C-
0491 C
0492          IS=1
0493          IF(KRS(9).EQ.1) IS=2
0494          IF(ITEST(IS,6).NE.3) GOTO 480
0495 C
0496 C
0497          PIN=PINTI
0498          PIN=SQRT((10.0)**(AKNEW))
0499 C
0500          DO 455 I=1,NPMN
0501          P(IS,I)=0.0
0502          IF(I.LE.30) G(IS,I)=0.0
0503 455      CONTINUE
0504 C
0505          IJ=0
0506          NLOOP=NSLF(IS)
0507          DO 460 I=1,NLOOP
0508          IJ=IJ+I
0509          IF(I.NE.IFP(IS,1).AND.I.NE.IFP(IS,2)) P(IS,IJ)=PIN

```



```

0510 C
0511 WRITE(IWIO,990)IS,I,IJ,P(IS,IJ)
0512 460 CONTINUE
0513 C
0514 480 CONTINUE
0515 IJK=IJK+1
0516 IF(IJK.LT.MULT) GOTO 301
0517 C
0518 C-----
0519 C-
0520 C- TIME TO DO CONTROL CALCULATION.
0521 C- NOTE, HAVE TO CHECK IF DOING BOTH SYSTEM OR JUST ONE
0522 C- FOR CASE WHERE 2 SYSTEM HAVE DIFFERENT SAMPLE RATE.
0523 C- SECOND MUST BE A MULTIPLE OF FIRST.
0524 C-
0525 C
0526 97 IJK2=IJK2+1
0527 NSYS=NSYSRD
0528 IF(IJK2.LT.MULT2) NSYS = 1
0529 IF(IJK2.GE. MULT2) IPIS(2)=IPIS(2)+1
0530 IF(IJK2. GE. MULT2) IJK2 = 0
0531 IJK=0
0532 M=M+1
0533 IF(M.GT.9999) M=1
0534 IPIS(1) = IPIS(1) + 1
0535 IF( IPIS(1) .GT. 30 ) IPIS(1) = 1
0536 IF( IPIS(2) .GT. 30 ) IPIS(2) = 1
0537 C
0538 C- GET INPUT MEASUREMENT.
0539 C- READING IN 5 POINTS, THEY ARE:
0540 C- POINT NO. DESCRIPTION
0541 C- 15 TOP COMPOSITION
0542 C- 10 DISTILLATE FLOW
0543 C- 13 REFLUX FLOW
0544 C- 9 STEAM FLOW
0545 C- 12 FEED
0546 C
0547 CALL AIRD(5,ICX,IDX,IERR)
0548 DO 96 KJI=1,5
0549 SADC(KJI)=FLOAT(IDX(KJI))
0550 C
0551 C
0552 96 CONTINUE
0553 YBTM=FLOAT(ICOM(ICOM(8)+1))*0.001
0554 YTOP=ADCM(1)*SADC(1)+ADCC(1)
0555 YN(1)=YTOP
0556 YN(2)=YBTM
0557 IDX(6)=YBTM
0558 C
0559 FEED = ADCM(1) * SADC(5) + ADCC(1)
0560 DIST = ADCM(1) * SADC(2) + ADCC(1)

```



```

0561      UDIS(1) = ADCM(1) * SADC(3) + ADCC(1)
0562      UDIS(2) = ADCM(1) * SADC(4) + ADCC(1)
0563      C
0564      C-----
0565      C-
0566      C-  FILTERING OF Y USING THE P-WEIGHT.
0567      C-  SWITCH 3 UP THEN WANT TO ADD DISTURBANCE INCREMENTALLY.
0568      C-
0569      C
0570      DO 159 IS=1,NSYS
0571      IP=IPIS(IS)
0572      DN(IS) = 0.0
0573      IF(ITEST(IS,3) .EQ. 4) DN(IS)=FEED-BD(IS)*KRS(13)
0574      DPR(IS)=(YN(IS)-YSTAR(IS))*IDPR
0575      SAVE(IS,1,IP)=0.0
0576      SAVE(IS,3,IP)=DN(IS)
0577      SAVE(IS,4,IP)=WN(IS)
0578      SAVE(IS,6,IP)=YN(IS)
0579      SAVE(IS,7,IP)=DPR(IS)
0580      159  CONTINUE
0581      C
0582      C
0583      DO 240 IS=1,NSYS
0584      IP=IPIS(IS)
0585      YPD = YN(IS)
0586      NLOOP=NP(IS)
0587      IF(NLOOP .LT. 2 ) GO TO 165
0588      C
0589      C-  YD = 1 / PD(Z) * YN
0590      C
0591      DO 160 J=2,NLOOP
0592      IC=IP-J+1
0593      IF(IC .LT. 1) IC=IC+30
0594      YPD = YPD - PD(IS,J)*SAVE(IS,2,IC)
0595      160  CONTINUE
0596      IF(PD(IS,1).LE.0.0) WRITE(IWIO,700)
0597      IF(PD(IS,1).LE.0.0) PD(IS,1)=1.0
0598      165  YPD = YPD/PD(IS,1)
0599      SAVE(IS,2,IP)=YPD
0600      C
0601      C-  YP = PN(Z) / PD(Z) * YN
0602      C
0603      YP = 0.0
0604      SAVE(IS,8,IP)=0.0
0605      C
0606      C
0607      NLOOP=NP(IS)
0608      DO 170 J=1,NLOOP
0609      IC=IP-J+1
0610      IF(IC .LT. 1) IC=IC+30
0611      YP = YP + PN(IS,J)*SAVE(IS,6,IC)

```



```

0612          YP = YP - PD(IS,J)*SAVE(IS,8,IC)
0613      170      CONTINUE
0614          YP = YP/PD(IS,1)
0615          SAVE(IS,8,IP)=YP
0616      C
0617      C-----
0618      C-
0619      C-  ACCUMULATE COST FUNCTION AND ERROR.
0620      C-
0621      C
0622          IC = IP - K1(IS) - 1
0623          IF(IC .LT. 1) IC=IC + 30
0624          PHI(IS)= YP - SAVE(IS,4,IC) + SAVE(IS,9,IC)
0625          ACLOSS(IS)=ACLOSS(IS) + PHI(IS)*PHI(IS)
0626          YERR(IS)=YN(IS) - YSTAR(IS)
0627      C
0628      C-----
0629      C-
0630      C-  CHECK WHICH IDENTIFICATION ALGORITHM.  AT PRESENT ONLY
0631      C-  LEAST SQUARE USING LEAST SQUARE ROOT.
0632      C-
0633      C
0634          IF(ITEST(IS,6) .EQ. 1) GO TO 200
0635          CALL IDENT2(PHI(IS),NSELF(IS),IS,PERERR(IS)
0636      *          ,X,PAR,P,G,ROW,IFP)
0637      200  CONTINUE
0638      C
0639      C-----
0640      C-
0641      C-  CALCULATING YSTAR AND X*THETA LESS THAT CONTRIBUTION OF THE
0642      C-  CONTROL ACTION IN THIS SCAN.
0643      C-
0644      C
0645          IJ = 0
0646          XP = 0.0
0647          YSTAR(IS)=0.0
0648      C
0649          DO 220 I=1,5
0650              NLOOP = N(IS,I)
0651              IF(NLOOP .EQ. 0) GO TO 220
0652              K=0
0653              ISTMP=IS
0654              ITEMP = I
0655              IF(I .EQ. 3) K=K3(IS)
0656              IF(I.EQ.3 .AND.IDPR.EQ.1) ITEMP=7
0657              IF(I .NE. 5) GO TO 205
0658                  ISTMP = 2
0659                  IF(IS .EQ. 2) ISTMP = 1
0660      C
0661      C
0662          ITEMP = 1

```



```

0663          K = K2(IS)
0664      205      DO 215 J=1,NLOOP
0665                IJ = IJ + 1
0666                IC = IP - J + 1 - K
0667                IF(IC .LT. 1) IC=IC+30
0668                XP=XP+PAR(IS,IJ)*SAVE(ISTMP,ITEMP,IC)
0669                IF(I.EQ.4) GO TO 210
0670                YSTAR(IS)=YSTAR(IS) +
0671      &          PAR(IS,IJ)*SAVE(ISTMP,ITEMP,IC)
0672      210      IC = IC - K1(IS)
0673                IF( IC .LT. 1) IC = IC + 30
0674                X(IS,IJ)=SAVE(ISTMP,ITEMP,IC)
0675      215      CONTINUE
0676      220      CONTINUE
0677      C
0678      C- CHECK TYPE OF VALVE ACTION.
0679      C
0680          IF(ITEST(IS,6).EQ.5) ACT(IS)=-ACT(IS)
0681          IF(ITYCT(IS) .NE. 3) GO TO 225
0682          XP =(YN(IS) - WN(IS)) * ABS(ACT(IS))/ACT(IS)
0683      225      CONTINUE
0684      C
0685      C-----
0686      C-
0687      C- CALCULATING NEW CONTROL ACTION.
0688      C-
0689      C-
0690          SAVE(IS,10,IP)=XP
0691          CO(IS)=0.0
0692          CA(IS)=QN(IS,1) + PAR(IS,1)*QD(IS,1)
0693          CB(IS)=0.0
0694          NLOOP=NQ(IS)
0695      C
0696          DO 230 J=1,NLOOP
0697                IC=IP-J+1
0698                IF( IC .LT. 1 ) IC=IC+30
0699                CO(IS)=CO(IS)-QD(IS,J)*SAVE(IS,10,IC)
0700                CO(IS)=CO(IS)-(QN(IS,J)+PAR(IS,1)*QD(IS,J))
0701      *          *SAVE(IS,1,IC)
0702      230      CONTINUE
0703      C
0704      C
0705          IF(N(IS,5) .EQ. 0) GO TO 240
0706          NS=NSSELF(IS)-N(IS,5)+1
0707          NLOOP=NQ(IS)
0708          CB(IS)=PAR(IS,NS)*QD(IS,1)
0709          IJ=NS
0710      C
0711          DO 235 J=1,NLOOP
0712                ISTMP=2
0713                IF(IS .EQ. 2) ISTMP=1

```



```

0714             IC=IP - J + 1
0715 C
0716 C
0717             IF(IC .LT. 1) IC=IC+30
0718             CO(IS)=CO(IS)-PAR(IS,NS)*QD(IS,J)
0719             *                               *SAVE(ISTHP,1,IC)
0720             235         CONTINUE
0721             240         CONTINUE
0722             DEN = CB(1)*CB(2) -CA(1)*CA(2)
0723             IF(DEN .EQ. 0.0) WRITE(IWIO,705)
0724             IF(DEN .EQ. 0.0) DEN =1.0
0725             UNT(1)=(CO(2)*CB(1)-CO(1)*CA(2))/DEN
0726             UNT(2)=(CO(1)*CB(2)-CO(2)*CA(1))/DEN
0727 C
0728 C-----
0729 C-
0730 C-  OBTAIN INDEX IP1 AND IP2 FOR PREVIOUS AND PREVIOUS,
0731 C-  PREVIOUS VALUE IN SAVE VECTOR.
0732 C-
0733 C
0734             DO 241 IS=1,NSYS
0735             IP=IPIS(IS)
0736             IP1(IS)=IP-1
0737             IF(IP1(IS).LT.1) IP1(IS)=30
0738             IP2(IS)=IP1(IS)-1
0739             IF(IP2(IS).LT.1) IP2(IS)=30
0740             241         CONTINUE
0741 C
0742 C-----
0743 C-
0744 C-  CALCULATING PID CONTROL ACTION.
0745 C-
0746 C
0747             DO 250 IS=1,NSYS
0748             IP=IPIS(IS)
0749             PCNT=(UTL(IS)-UBL(IS))*IRAMP(IS)/100.0
0750             UTOP      =UN(IS)+PCNT
0751             UBOT      =UN(IS)-PCNT
0752             ERR2(IS)= ERR1(IS)
0753             ERR1(IS)= ER1(IS)
0754             ER1(IS) = UN(IS) - YN(IS)
0755 C
0756 C-  UPDATING SIAE COUNTER.
0757 C
0758             IF(ITEST(IS,6).EQ.6) SIAE(IS)=0.0
0759             SIAE(IS)= SIAE(IS) + ABS(ER1(IS))
0760             UPROP(IS)=AKPS(IS)*ER1(IS)
0761             IF(ITEST(IS,8).LE.2) GOTO 247
0762             IF(ITEST(IS,8).EQ.4.AND.AKIS(IS).NE.0.0) GOTO 247
0763             IF(ITEST(IS,8).EQ.4) GOTO 248
0764 C

```



```

0765      UDER(IS)=AKDS(IS)*(ERR1(IS)-ER1(IS))
0766      UINT(IS)=UINT(IS)+AKIS(IS)*ER1(IS)
0767      UPID(IS)=UPROP(IS)+UINT(IS)+UDER(IS)
0768      GOTO 246
0769  C
0770  C
0771  C
0772  247      UINT(IS)=UN(IS)-UPROP(IS)
0773          IF(ITEST(IS,8).EQ.4) GOTO 248
0774  C
0775  C-----
0776  C-
0777  C- INCREMENTAL CONTROL EQUATION
0778  C-
0779  C
0780  245      UPROP(IS)= ER1(IS)-ERR1(IS)
0781          ID1=IP1(IS)
0782          ID2=IP2(IS)
0783          UDER(IS)=AKDS(IS)*(YN(IS)-2.*SAVE(IS,6,ID1)
0784  *          +SAVE(IS,6,ID2))
0785          UDINT(IS)=AKIS(IS)*ER1(IS)
0786          USINT(IS)=USINT(IS)+UDINT(IS)
0787          DELTA=UPROP(IS)+UDINT(IS)+UDER(IS)
0788          UPID(IS)=UPID(IS)+DELTA*AKPS(IS)
0789          IF(ITEST(IS,8) .EQ. 4) UPID(IS) = USS(IS)
0790  C
0791  246      IF(ITEST(IS,8).EQ.0) UN(IS)=UPID(IS)
0792          IF(ITEST(IS,8).EQ.3) UN(IS)=UPID(IS)
0793          IF(ITEST(IS,8).NE.0.AND.ITEST(IS,8).NE.3)
0794  *          UPID(IS) = UN(IS)
0795          IF(ITEST(IS,8).EQ.2) UN(IS)=UNT(IS)+USINT(IS)
0796          IF(ITEST(IS,8).EQ.1) UN(IS)=UNT(IS)
0797  248      IF(ITEST(IS,8).EQ.4) UN(IS)=USS(IS)
0798  C
0799  C-----
0800  C-
0801  C- LIMIT CHECKING.
0802  C-
0803  C
0804          IF(ITEST(IS,1) .GE. 1) WRITE(IWID,630) UN(IS)
0805          IF(UN(IS) .GT. UTOP      ) UN(IS)=UTOP
0806          IF(UPID(IS) .GT. UTOP      ) UPID(IS)=UTOP
0807          IF(UN(IS) .LT. UBOT      ) UN(IS)=UBOT
0808          IF(UPID(IS) .LT. UBOT      ) UPID(IS)=UBOT
0809  C
0810  C- CHECK RESET WINDUP.
0811  C
0812          IF(UN(IS).LT.UTL(IS)) GOTO 249
0813          UINT(IS)=UINT(IS)+(UN(IS)-UTL(IS))
0814          USINT(IS)=USINT(IS)+(UN(IS)-UTL(IS))
0815          UN(IS)=UTL(IS)

```



```

0816 249 IF(UN(IS).GT.UBL(IS)) GO10 942
0817      UINT(IS)=UINT(IS)+(UN(IS)-UBL(IS))
0818      USINT(IS)=USINT(IS)+(UN(IS)-UBL(IS))
0819      UN(IS)=UBL(IS)
0820 942 CONTINUE
0821 C
0822 C-----
0823 C-
0824 C- CALCULATING THE NEW YSTAR I.E. ESTIMATE OF Y
0825 C
0826 C
0827 C-
0828      ISTMP = 2
0829      IF(IS .EQ. 2) ISTMP = 1
0830      NS = NSELF(IS) - N(IS,5) + 1
0831      YSTAR(IS)=YSTAR(IS)+PAR(IS,1)*UN(IS)
0832      *          +PAR(IS,NS)*UN(ISTMP)
0833      SAVE(IS,1,IP)=UN(IS)
0834 C
0835 C- IF SWITCH 3 IS UP, WANT TO IDENTIFY ONLY. READ IN THE
0836 C- CONTROL
0837 C
0838      IF(KRS(14).EQ.1) SAVE(IS,1,IP) = UDIS(IS)
0839      IC=IP-K1(IS)
0840      IF(IC .LT. 1) IC=IC+30
0841      X(IS,1)=SAVE(IS,1,IC)
0842 250 CONTINUE
0843 C
0844 C-----
0845 C-
0846 C- SENDING OUTPUT
0847 C-
0848 C
0849 C
0850      DO 251 IS=1,NSYS
0851          USEND(IS) = UN(IS)
0852          IF(ACT(IS) .LT. 0) USEND(IS) = ABS(ACT(IS))-UN(IS)
0853          USEND(IS) = USEND(IS)**2 * DACM(IS) + DACC(IS)
0854          IOUT = IFIX(USEND(IS))
0855          IF(KRS(16) .EQ. 1) CALL OUT(IOUTCH(IS),IOUT,IERR)
0856          IF(KRS(16).EQ.0) WRITE(IWIO,1234)
0857 1234 FORMAT(' NOT SENDING OUTPUT')
0858 251 CONTINUE
0859 C
0860 C*****
0861 C*
0862 C* SECTION TO PRINT OUT THE RESULTS.
0863 C*
0864 C*****
0865 C
0866      DO 299 IS=1,NSYS

```



```

0867      IP=IPIS(IS)
0868      ICH=ITEST(IS,8)+1
0869      IF(ITEST(IS,1).EQ.0) GO TO 274
0870      IF(ITEST(IS,1).LT.5)
0871      *                               WRITE(IWIO,615) CHAR(ICH),IS,
0872      *                               M,WN(IS),YN(IS),YSTAR(IS),DN(IS),
0873      *                               UN(IS),USEND(IS),PHI(IS),SIAE(IS)
0874  C
0875  272      IF(ITEST(IS,1).LT.3 .OR. ITEST(IS,1).GT.4) GO TO 273
0876      NPAR=NSELF(IS)
0877      WRITE(IWIO,625)(IS,J,PAR(IS,J),G(IS,J),J=1,NPAR)
0878  C
0879  273      IF(ITEST(IS,1).EQ.4)
0880  C
0881  C
0882      *                               CALL PWRITE(ITYID(IS),
0883      *                               NSELF(IS),IS,IWIO,1,X,PAR,P,G,ROW,IFP)
0884  274      CONTINUE
0885  C
0886  C-----
0887  C-
0888  C- UPDATE THE MEASUREMENT VECTOR, I.E. ADD EFFECT OF SETPOINT
0889  C- AND INTERACTIVE TERMS.
0890  C-
0891  C
0892      IF(N(IS,5).EQ.0) GO TO 255
0893      NS=NSELF(IS)-N(IS,5)+1
0894      IC=IP-K2(IS)-K1(IS)
0895      IF(IC.LT.1) IC=IC+30
0896      ISTMP=2
0897      IF(IS.EQ.2)ISTMP=1
0898      X(IS,NS)=SAVE(ISTMP,1,IC)
0899  255      CONTINUE
0900  C
0901  C-----
0902  C-
0903  C- CALCULATED Q-FILTERED VALUE OF U(T).
0904  C-
0905  C
0906      UQ = 0.0
0907      NLOOP=NQ(IS)
0908      SAVE(IS,9,IP)=0.0
0909      IF(ITYCT(IS) .EQ. 3)      GO TO 265
0910      DO 260 J=1,NLOOP
0911      IC=IP-J+1
0912      IF(IC .LT. 1)      IC = IC + 30
0913      UQ = UQ - QD(IS,J)*SAVE(IS,9,IC)
0914      UQ = UQ + QN(IS,J)*SAVE(IS,1,IC)
0915  260      CONTINUE
0916  265      IF(QD(IS,1).EQ.0.0) WRITE(IWIO,710)
0917      IF(QD(IS,1).EQ.0.0) QD(IS,1)=1.0

```



```

0918             SAVE(IS,9,IP)=UQ/QD(IS,1)
0919 C
0920 C* ESTIMATING W-OLD
0921 C
0922     297     IC = IP -K1(IS)
0923             IF(IC .LT. 1) IC=IC+30
0924             WOLD(IS) = SAVE(IS,4,IC)
0925             NS = NSELF(IS) - N(IS,5) - N(IS,4) + 1
0926             NF = NSELF(IS) - N(IS,5)
0927 C             IF(ITYCT(IS) .EQ. 1) GO TO 290
0928             IF(ITYCT(IS) .EQ. 3) GO TO 290
0929             DO 298 I=NS,NF
0930                 IC=IP - I - K1(IS) + NS
0931                 IF(IC .LT. 1) IC=IC+30
0932                 X(IS,I)=X(IS,I) - SAVE(IS,9,IC)
0933     298             CONTINUE
0934     290             CONTINUE
0935 C
0936 C
0937     299 CONTINUE
0938 C
0939 C
0940 C-----
0941 C-
0942 C- CHECK IF IT IS TIME TO STORE INFORMATION TO DISK.
0943 C-
0944             ICNT=ICNT+1
0945             IF(ICNT.LT.IDISK) GOTO 301
0946 C
0947 C-----
0948 C-
0949 C- TRANSFER INFORMATION TO DISK.
0950 C-
0951 C
0952             ICNT=0
0953             IDISK=IDMP
0954 C
0955             DO 305 IS=1,NSYS
0956             IF(ITEST(IS,7).EQ.4) GOTO 999
0957             IF(ITEST(IS,7).EQ.7) IDMP=IFIX(AKW(IS))
0958             IF(ITEST(IS,7).LE.1) GOTO 307
0959             BUF(1)=FLOAT(ICH)
0960             BUF(2)=FLOAT(IS)
0961             BUF(3)=FLOAT(H)
0962             BUF(4)=WN(IS)
0963             BUF(5)=YN(IS)
0964             BUF(6)=UN(IS)
0965             BUF(7)=PHI(IS)
0966             BUF(8)=SIAE(IS)
0967             BUF(9)=YTOP
0968             BUF(10)=UDIS(1)

```



```

0969      BUF(11)=YBTH
0970      BUF(12)=FEED
0971      BUF(13)=UDIS(2)
0972      BUF(14)=DIST
0973      BUF(15)=YSTAR(1)
0974      BUF(16)=YSTAR(2)
0975      C
0976      C-----
0977      C-
0978      C-  BUF(15) ---- BUF(85)
0979      C-
0980      C
0981      C
0982      NPAR=NSELF(IS)
0983      NPTS = 2 * (16 + 2*NPAR)
0984      DO 302 I=1,NPAR
0985      302  BUF(16+I)=PAR(IS,I)
0986      C
0987      DO 303 I=1,NPAR
0988      303  BUF(16+NPAR+I)=G(IS,I)
0989      C
0990      C
0991      C
0992      C
0993      C  IF(NPAR.GE.10) GOTO 308
0994      C  NPE=(NPAR**2+NPAR)/2
0995      C  DO 304 I=1,NPE
0996      C304  BUF(14+2*NPAR+I)=P(IS,I)
0997      C
0998      308  IF(IS.EQ.2) GOTO 306
0999      CALL WRITF(IDCBI,IERR,BUF,NPTS)
1000      GOTO 307
1001      306  CALL WRITF(IDCBI,IERR,BUF,NPTS)
1002      307  CONTINUE
1003      C
1004      305  CONTINUE
1005      C
1006      C*  CALL TIME TO CHECK HOW LONG PROGRAM SHOULD BE SUSPENDED
1007      C*  BEFORE NEXT SCAN.  NOTE, EXCEPT FOR INITIAL SCAN, THE
1008      C*  START TIME OF THE NEXT SCAN IS CALCULATED FROM END OF
1009      C*  THIS SCAN + TIME IN WAIT.  IF THE PROGRAM BECOMES
1010      C*  SUSPENDED FOR LONGER THAN SPECIFIED TIME (DUE TO LOAD-
1011      C*  ING) THE NEXT SUSPEND WILL BE SHORTENED.
1012      C
1013      301  CONTINUE
1014      CALL EXEC(11,ITIME)
1015      ITM2=100*ITIME(2)+ITIME(1)
1016      IDIF=ITM2-ITM1
1017      312  IF(IDIF.GE.0) GO TO 315
1018      IDIF=IDIF+6000
1019      GO TO 312

```



```

1020 C
1021 C* CHECK IF CONTROL CALCULATION WITHIN THE SCAN TIME.
1022 C
1023 315 IF(IDIF.LE.ISCAN) GO TO 320
1024 WRITE(IWIO,900) IDIF
1025 IDIF=IDIF-ISCAN
1026 ITM1=ITM1+ISCAN
1027 IJK=IJK+1
1028 IF(IJK.GE.MULT) GO TO 97
1029 GO TO 315
1030 C
1031 320 IWAIT=ISCAN-IDIF
1032 ITM1=ITM2+IWAIT
1033 CALL WAIT(IWAIT,0,IERR)
1034 GO TO 95
1035 C
1036 C-----
1037 C-
1038 C- CLOSE FILES AND EXIT
1039 C-
1040 C
1041 999 CALL CLOSE(IDC1,IERR)
1042 CALL CLOSE(IDC2,IERR)
1043 C
1044 IF(ITEST(IS,8) .EQ. 6) GO TO 5
1045 C
1046 C
1047 IF(ITEST(IS,7) .EQ. 4) GO TO 91
1048 STOP
1049 C
1050 C*****
1051 C*
1052 C* FORMAT STATEMENTS
1053 C*
1054 C*****
1055 C
1056 600 FORMAT(' AK-',I1,'=',G12.5,' P',F8.2,' I',F8.2,
1057 &' D',F8.2,' W',F8.2,' D',F8.2,/,
1058 &' PID =',F8.2,' I',F8.2,' D',F8.2)
1059 605 FORMAT(' RESET Q-FILTER IS=',I1,/,
1060 &' 1X,' AKP=',F8.3,' AKI=',F8.3,' AKD=',F8.3,/,
1061 &' QD(',I1,',',I1,')=',G12.5,1X,G12.5,1X,G12.5,1X,
1062 &G12.5,/, ' QN(-,-)=',G12.5,1X,G12.5,1X,G12.5,1X,G12.5,1X)
1063 606 FORMAT(' T-P-FILTER =',G12.5)
1064 607 FORMAT(' RESET P-FILTER T-P=',G12.5,/,
1065 &' PD(',I1,',',I1,')=',G12.5,1X,G12.5,1X,G12.5,1X,
1066 &G12.5,/, ' PN(-,-)=',G12.5,1X,G12.5,1X,G12.5,1X,G12.5)
1067 610 FORMAT(' ITEST=',8(I2,1X),5X,8(I2,1X) )
1068 615 FORMAT(10X,' ITR',2X,' SETPT',2X,' MEAS.',3X,' YSTAR',5X,
1069 *' DN',6X,' UN',5X,' OUT',6X,, ' PHI',5X,' SIAE',/,
1070 * 1X,A4,' M-',I1,'=',I5,1X,F5.2,1X,F7.3,1X,F7.3,1X,

```



```

1071      &F7.2,1X,F7.3,1X,F7.1,1X,G8.3,1X,G8.3)
1072 625 FORMAT(1X,'PAR(',I1,',',I3,')=',G12.5,' K-G=',G12.5)
1073 630 FORMAT(1X,'CAL. OUTPUT = ',G10.5)
1074 700 FORMAT(' PD IS ZERO, RESET TO 1.0 ')
1075 705 FORMAT(' CA IS ZERO, RESET TO 1.0 ')
1076 710 FORMAT(' QD IS ZERO, RESET TO 1.0 ')
1077 800 FORMAT(2X,'ENTER NSYS,IDMP,IDPR: ', '_')
1078 805 FORMAT(2I7)
1079 810 FORMAT(1X,5(G12.5,1X))
1080 815 FORMAT(1X,'TP-PR ',F7.2,' DS-FL ',F7.2,' RF-FL ',F7.2,
1081      &      'ST-FL ',F7.2,' FE-FL ',F7.2,' BT-PR ',F7.2)
1082 820 FORMAT(2X,'INPUT DATA DUMP FILES.....MUST EXIST',
1083      *      ' ALREADY',/,
1084      &      2X,'ITEST(7) MUST BE 2 TO DUMP !!!!')
1085 825 FORMAT(3A2,I3)
1086 900 FORMAT(2X,' EXCEED TIME SCAN, TIME NEEDED = ',I10)
1087 990 FORMAT(' IS=',I2,' LOOP=',I7,' P(',I4,')=',G12.5)
1088      END

```

:


```

0001  FTN4
0002      SUBROUTINE ZTRAN(ITYP,TS,GAIN,T,A,B,NA,NB)
0003      DIMENSION T(3),A(20),B(20)
0004      DO 10 I=1,20
0005          A(I)=0.0
0006          B(I)=0.0
0007      10  CONTINUE
0008      NA=1
0009      NB=1
0010      A(1)=1.0
0011      B(1)=0.0
0012      IF(ITYP .LE. 0 .OR. ITYP .GT. 5) GO TO 40
0013      GO TO( 15 , 20 , 25 , 30 , 35) , ITYP
0014  CC      FIRST ORDER LAG      1/(TS+1) .....
0015  C
0016      15      NB=1
0017              NA=2
0018              B(1)=GAIN*(1.0-EXP(-TS/T(1)) )
0019              A(2)=-EXP(-TS/T(1))
0020              GO TO 40
0021  C
0022  C      SECOND ORDER  1/(S)(S+1).....
0023  C
0024      20      NB=2
0025              NA=3
0026              B(1)=GAIN*(TS-T(1)*(1.-EXP(-TS/T(1))) )
0027              B(2)=GAIN*( -EXP(-TS/T(1))*TS + T(1)*(1.-EXP(-TS/T(1))) )
0028              A(2)=-(1.0+EXP(-TS/T(1)))
0029              A(3)=-EXP(-TS/T(1))
0030              GO TO 40
0031  C
0032  C      SECOND ORDER  1/(ST1+1)(ST2+1)
0033  C
0034      25      AA=EXP(-TS/T(1))
0035              BB=EXP(-TS/T(2))
0036              NB=2
0037              NA=3
0038              B(1)=GAIN*( (BB/T(2)-AA/T(1))/(1./T(2)-1./T(1))-(AA+BB)+1
0039              B(2)=GAIN*(AA*BB-(BB/T(2)-AA/T(1))/(1./T(2)-1./T(1)) )
0040              A(2)=-(BB+AA)
0041              A(3)=AA*BB
0042              GO TO 40
0043  C
0044  C      SECOND ORDER      1/(TS + 1)(TS + 1)
0045  C
0046      30      NB=2
0047              NA=3
0048              AA=EXP(-TS/T(1))
0049              B(1)=GAIN*( -(1.+TS/T(1))*AA+1.0)
0050              B(2)=GAIN*(AA*AA - (1. - TS/T(1))*AA)
0051              A(2)=-2.*AA

```



```

0052      A(3)=AA*AA
0053      GO TO 40
0054 C
0055 C      SECOND ORDER      1/(T1S*S + T2S + 1).....
0056 C
0057      35      AA = T(2)/(2.*T(1))
0058      BB = 1/T(1) - (T(2)*T(2))/(T(1)*2.)*2
0059      IF(BB.LT.0.0) GO TO 40
0060      BB=SQRT(BB)
0061      B(1)=EXP(-AA*TS)*(-COS(BB*TS)-AA/BB*SIN(BB*TS))+1.0
0062      B(2)=EXP(-2.*AA*TS)-EXP(-AA*TS)*(COS(BB*TS)-AA
0063      &      /BB*SIN(BB*TS))
0064      A(1)=1.0
0065      A(2)=-2.*EXP(-AA*TS)*COS(BB*TS)
0066      A(3)=EXP(-2.*AA*TS)
0067      A(3)=EXP(-2.*AA*TS)
0068      B(1)=GAIN*B(1)/(T(1)*(BB*BB+AA*AA))
0069      B(2)=GAIN*B(2)/(T(1)*(BB*BB+AA*AA))
0070      NB=2
0071      NA=3
0072      40 RETURN
0073      END
:
```

```

0179      SUBROUTINE RSW(IRSW)
0180      DIMENSION IRSW(16)
0181      CALL DI(1,1,ID,IERR)
0182      IRSW(16)=0
0183      IF(ID .LT. 0) IRSW(16)=1
0184      IF(ID .LT. 0) ID = ID + 32766 + 2
0185      IF(IERR .NE. 1) GO TO 999
0186      DO 100 I = 2,16
0187      IJ = 16 - I
0188      IRSW(IJ+1)=0
0189      IDT = ID / (2**IJ)
0190      IF(IDT .EQ. 1) IRSW(IJ+1)=1
0191      IF(IDT .EQ. 1) ID = ID - 2**IJ
0192      100 CONTINUE
0193      DO 200 I = 1,8
0194      ITEMP      =      IRSW(17-I)
0195      IRSW(17-I)=      IRSW(I)
0196      IRSW(I)      =      ITEMP
0197      200 CONTINUE
0198      999 RETURN
0199      END
```



```

0046      SUBROUTINE IDENT2(YNEW,N,IS,PERER,X,A,P,G,ROW,IFP)
0047      DIMENSION X(2,30),A(2,30),P(2,465),G(2,30),ROW(2,3),IFP(2,2)
0048      C
0049      C.....
0050      C
0051      C
0052      C      RECURSIVE SQUARE ROOT ALGORITHM.....
0053      C
0054      C
0055      C.....
0056      C
0057          PERER = YNEW
0058          DO 10 I=1,N
0059              PERER = PERER - X(IS,I)*A(IS,I)
0060      10  CONTINUE
0061      C
0062          GAMA = ROW(IS,1)
0063          GAMA2= ROW(IS,1)*ROW(IS,1)
0064          IJ=0
0065          JI=0
0066          DO 30 J=1,N
0067              PX = 0.0
0068              J1 = J - 1
0069              DO 15 I=1,J
0070                  JI = JI + 1
0071                  PX = PX + P(IS,JI)*X(IS,I)
0072      15  CONTINUE
0073              ALFA = GAMA/ROW(IS,2)
0074              BETA = PX/GAMA2
0075              GAMA2 = GAMA2 + PX*PX
0076              GAMA = SQRT(GAMA2)
0077              ALFA = ALFA/GAMA
0078              G(IS,J) = P(IS,JI)*PX
0079              P(IS,JI) = ALFA*P(IS,JI)
0080              IF(J1 .EQ. 0) GO TO 25
0081                  DO 20 I=1,J1
0082                      IJ = IJ + 1
0083                      PQP = P(IS,IJ)
0084                      P(IS,IJ)=ALFA*(PQP-BETA*G(IS,I) )
0085                      G(IS,I)=G(IS,I) + PQP*PX
0086      20  CONTINUE
0087      25  CONTINUE
0088              IJ = IJ + 1
0089      30  CONTINUE
0090      C
0091          PERER = PERER
0092          DO 35 I=1,N
0093              G(IS,I)=G(IS,I)/GAMA2
0094              A(IS,I) = A(IS,I) + G(IS,I)*PERER
0095      35  CONTINUE
0096      C
0097          RETURN
0098          END

```



```

0099      SUBROUTINE PWRITE(ITYPE,N,IS,IWIO,IPRNT,X,A,P,G,ROW,IFP)
0100      DIMENSION D(30) , CHAR(5)
0101      DIMENSION X(2,30),A(2,30),P(2,465),G(2,30),ROW(2,3),IFP(2,2)
0102      C.....
0103      C
0104      C      PRINTING ROUTINE .....
0105      C
0106      C
0107      C
0108      C.....
0109      C
0110      C
0111      DATA CHAR/4HRLS ,4HRSR ,4HUDU ,4HRL ,4H+---/
0112      JS = 0
0113      IF(ITYPE.LT.1 .OR. ITYPE .GT. 3) GO TO 65
0114      IF(IPRNT.EQ.1) WRITE(IWIO,70)CHAR(ITYPE)
0115      DO 60 I=1,N
0116          DO 10 L=1,N
0117      10      D(L)=0.0
0118          GO TO ( 15 , 25 , 40 , 65),ITYPE
0119      C
0120      C----- RLS -----
0121      C
0122      15      DO 20 J=I,N
0123          IJ= J*(J-1)/2 + I
0124          IF(J .LT. I) IJ= I*(I-1)/2 + J
0125          D(J-I+1)=P(IS,IJ)
0126      20      CONTINUE
0127          GO TO 55
0128      C
0129      C-----RSR -----
0130      C
0131      25      DO 35 J=I,N
0132          KDO = N - J + 1
0133          J1 = J
0134          DO 30 K=1,KDO
0135              J2 = J1*(J1-1)/2 + I
0136              J3 = J1*(J1-1)/2 + J
0137              D(J-I+1)=D(J-I+1) + P(IS,J2)*P(IS,J3)
0138              J1=J1+1
0139      30      CONTINUE
0140      35      CONTINUE
0141          GO TO 55
0142      C
0143      C----- U - D - U -----
0144      C
0145      40      J3 = JS
0146          JDO= N - I + 1
0147          DO 50 J=1,JDO
0148              D(J)=0.0

```



```

0149          SUM =1.0
0150          KDO = JDO - J + 1
0151          DO 45 K=1,KDO
0152              J1 = JS + (J-1) + K
0153              J3 = J3 + 1
0154              J2 = J2 + KDO - K + 2
0155              IF(K .EQ. 1) J2 = J3
0156              IF(J.GT.1) SUM = SUM * P(IS,J1)
0157              IF(K.GT.1) SUM = SUM * P(IS,J3)
0158              SUM = SUM * P(IS,J2)
0159              D(J)= D(J)+SUM
0160          45      CONTINUE
0161          50      CONTINUE
0162              JS = J1
0163      C
0164      C
0165      C----- PRINT SECTION -----
0166      C
0167          55      ND= N - I + 1
0168              WRITE(IWIO, 80)ND,(D(L),L=1,ND)
0169              IF(ND.GT.5)ND=5
0170              IF(IPRNT.EQ.1) WRITE(IWIO, 75)(CHAR(5),L=1,ND)
0171          60      CONTINUE
0172          65      RETURN
0173          70      FORMAT(' +- ',A4,'--+- DIAGNAL --+',/,
0174              &          ' +-----+', '-----+')
0175          75      FORMAT(' +-----',5(A4,9H----- ))
0176          80      FORMAT(' I P',I3,' I',20(G12.5,'I',G12.5,'I',G12.5,'I',
0177              &          G12.5,'I',G12.5,'I',/, ' I CONT I' ) )
0178      END
:
```



```

0001  FTN4
0002  PROGRAM HSET ( ) , THIS PROGRAMME GENERATES SELF-TUNER DATAFI
0003  DIMENSION D(6,85) , N(10) , AIN(10) ,RTYPE(6)
0004  DIMENSION IDCB(144) , BUF(45) , IFILE(3)
0005  DIMENSION PRIDNT(5),PRITYP(3)
0006  DIMENSION T(3) , A(20) , B(20)
0007  DATA PRIDNT/4HNONE,4HRLS ,4HRSR ,4HU-D ,4HRL /
0008  DATA RTYPE/4HTOP ,4HTOP ,4HTOP ,4HBOT ,4HBOT ,4HBOT /
0009  DATA PRITYP/4HST ,4HSTPI,4HPID /
0010  DATA KC,KH,KS,KR/1HC,1HH,1HS,1HR/
0011  DATA KTWO/1H2/
0012  DATA ILP/2HLP/
0013  C
0014  C
0015  DATA ILEN,IFILE/170,2H&H,2HDA,2HTA/
0016  C
0017  C    ...< I/O ROUTINE >.....
0018  C
0019  IW=LOGLU(IDUMMY)
0020  IR=IW
0021  CALL OPEN(IDCB,IERR,IFILE,0,77,136,144)
0022  IF(IERR .GE. 0) GO TO 5
0023  CALL FMGER(IERR)
0024  STOP 4
0025  C
0026  C
0027  5  WRITE(IW,320)
0028  READ(IW,*) IFLG
0029  320 FORMAT(' ENTER 1 IF INITIALIZING FILE: ','_')
0030  IF(IFLG.EQ.1) GO TO 15
0031  DO 10 J=1,6
0032  CALL READF(IDCB,IERR,BUF,80,ILEN)
0033  IF(IERR .GE. 0) GO TO 6
0034  CALL FMGER(IERR)
0035  STOP 4
0036  6  DO 800 JJ = 1,40
0037      D(J,JJ) = BUF(JJ)
0038  800 CONTINUE
0039  C
0040  CALL READF(IDCB,IERR,BUF,90,ILEN)
0041  DO 802 JJ = 41,85
0042  IDUM = JJ - 40
0043  D(J,JJ) = BUF(IDUM)
0044  802 CONTINUE
0045  10 CONTINUE
0046  15 WRITE(IW,125)
0047  IG=0
0048  READ (IR,310)ICHR,ICHARS,ILOG
0049  IF(ICHARS.EQ.KTWO) IG=1
0050  IF(ICHR .EQ. KR ) GO TO 90
0051  IF(ICHR.EQ.KC .OR.ICHR.EQ.KM .OR.ICHR.EQ.KS )

```



```

0052      * GO TO 20
0053      GO TO 15
0054      20  ISYS=1 + IG*3
0055      IF(ICHAR .EQ. KS      )  GO TO 35
0056      IF(ICHAR .EQ. KC      )  GO TO 45
0057      C
0058      C.....MODEL SECTION.....
0059      C
0060      IF(ILOG.EQ.ILP) IW=6
0061      NTYPE=IFIX(D(ISYS,1))
0062      WRITE(IW,1500) RTYPE(ISYS)
0063      WRITE(IW,150)NTYPE
0064      IF(NTYPE .GT. 0) GO TO 25
0065      C
0066      C..... Z - TRANSFER MODEL SIMULATION .....
0067      C
0068      NF =IFIX(D(ISYS,10))
0069      NFP=NF+11
0070      WRITE(IW,155)NF,(D(ISYS,J),J=11,NFP)
0071      WRITE(IW,120)
0072      NF=IFIX(D(ISYS,20))
0073      NFP=NF+21
0074      WRITE(IW,160)NF,(D(ISYS,J),J=21,NFP)
0075      WRITE(IW,120)
0076      NF=IFIX(D(ISYS,30))
0077      NFP=NF+31
0078      WRITE(IW,165)NF,(D(ISYS,J),J=31,NFP)
0079      WRITE(IW,120)
0080      NF=IFIX(D(ISYS,40))
0081      NFP=NF+41
0082      WRITE(IW,170)NF,(D(ISYS,J),J=41,NFP)
0083      WRITE(IW,120)
0084      NF=IFIX(D(ISYS,50))
0085      NFP=NF+51
0086      WRITE(IW,175)NF,(D(ISYS,J),J=51,NFP)
0087      WRITE(IW,120)
0088      GO TO 30
0089      C
0090      C.....RK - SIMULATION.....
0091      C
0092      25  WRITE(IW,180)(D(ISYS,J),J=10,14)
0093      WRITE(IW,120)
0094      WRITE(IW,185)(D(ISYS,J),J=20,24)
0095      WRITE(IW,120)
0096      WRITE(IW,190)(D(ISYS,J),J=30,34)
0097      WRITE(IW,120)
0098      WRITE(IW,195)(D(ISYS,J),J=40,44)
0099      WRITE(IW,120)
0100      30  WRITE(IW,200)(D(ISYS,J),J=2,6)
0101      DELAY = D(ISYS,9)-D(ISYS,7)
0102      WRITE(IW,205)(D(ISYS,J),J=7,9),DELAY

```



```

0103      WRITE(IW,210)(D(ISYS,J),J=80,83)
0104      WRITE(IW,120)
0105      IW=IR
0106      GO TO 70
0107  C
0108  C.....SYSTEM SECTION.....
0109  C
0110      35  ISYS=2 + IG*3
0111      DO 40 J=1,9
0112          N(J)=IFIX(D(ISYS,J))
0113      40  CONTINUE
0114      IF(ILOG.EQ.ILP) IW=6
0115      NFIX=N(1)+N(2)+N(3)+1
0116      WRITE(IW,2150) RTYPE(ISYS)
0117      WRITE(IW,215)(N(J),J=1,8),NFIX,N(9)
0118      WRITE(IW,120)
0119      NF=N(1)+10
0120      WRITE(IW,220)(D(ISYS,J),J=10,NF)
0121      NF=N(2)+20
0122      WRITE(IW,225)(D(ISYS,J),J=20,NF)
0123      NF=N(3)+30
0124      WRITE(IW,230)(D(ISYS,J),J=30,NF)
0125      NF=N(4)+40
0126      WRITE(IW,235)(D(ISYS,J),J=40,NF)
0127      NF=N(5)+50
0128      WRITE(IW,240)(D(ISYS,J),J=50,NF)
0129      WRITE(IW,120)
0130      NF=IFIX(D(ISYS,60))
0131      NFP = NF+1
0132      IF(NFP .LT. 1 .OR. NFP .GT. 5) NFP = 1
0133      N1=IFIX(D(ISYS,64))
0134      N2=IFIX(D(ISYS,65))
0135      STD=2.75*SQRT( ABS( (D(ISYS-1,5)**2+0.00001)/30.0 ))
0136      WRITE(IW,245)NF,PRIDNT(NFP),(D(ISYS,J),J=61,63),N1,N2,D(ISY
0137      & ,D(ISYS,67),STD
0138      WRITE(IW,120)
0139      WRITE(IW,250)(D(ISYS,J),J=70,75)
0140      WRITE(IW,120)
0141      WRITE(IW,255)(D(ISYS,J),J=80,85)
0142      WRITE(IW,120)
0143      IW=IR
0144      GO TO 70
0145  C
0146  C.....CONTROLLER SECTION.....
0147  C
0148      45  ISYS=3 + IG*3
0149      N1=IFIX(D(ISYS,1))
0150      IF(N1.LT.1 .OR. N1.GT.3) N1=1
0151      N2=IFIX(D(ISYS,5))
0152      NQ=IFIX(D(ISYS,10))
0153      NP=IFIX(D(ISYS,20))

```



```

0154      N3=IFIX(D(ISYS,30))
0155      NR=IFIX(D(ISYS,40))
0156      PCNT=(D(ISYS,3)-D(ISYS,2))*D(ISYS,4)/100.0
0157      IF(ILOG.EQ.ILP) IW=6
0158      WRITE(IW,2600) RTYPE(ISYS)
0159      WRITE(IW,260)N1,PRITYP(N1),(D(ISYS,J),J=2,4),PCNT
0160  C
0161  C.....Q-FILTER.....
0162  C
0163      WRITE(IW,265)N2,D(ISYS,6)
0164      IF(N2 .EQ. 0)      GO TO 55
0165      WRITE(IW,270)(D(ISYS,J),J=7,9)
0166      WRITE(IW,120)
0167  C
0168  C..... ESTIMATION OF EQUILANT Z-TRANSFORM.....
0169  C
0170      NQ = 1
0171      D(ISYS,15)=1.0
0172      D(ISYS,16)=0.0
0173      IF(D(ISYS,7) .GT. 0.0) GO TO 50
0174      IF(D(ISYS,8) .EQ. 0.0) GO TO 55
0175  C
0176  C.....INTEGRAL ACTION ONLY .....
0177  C
0178      NQ=2
0179      D(ISYS,11)=D(ISYS,8)
0180      D(ISYS,12)=0.0
0181      D(ISYS,16)=-1.0
0182      GO TO 55
0183  50  NQ=1
0184      D(ISYS,11)=D(ISYS,7)+D(ISYS,8)
0185      D(ISYS,12)=0.0
0186      D(ISYS,13)=D(ISYS,9)
0187      D(ISYS,16)=0.0
0188      IF(D(ISYS,8) .LE. 0.0) GO TO 55
0189      NQ=2
0190      D(ISYS,12)=-D(ISYS,7)-2.*D(ISYS,9)
0191      D(ISYS,16)=-1.0
0192      IF(D(ISYS,9) .GT. 0.0) NQ=3
0193  55  WRITE(IW,275)NQ,(D(ISYS,J),J=11,14)
0194      WRITE(IW,280) (D(ISYS,J),J=15,18)
0195      D(ISYS,10)=NQ
0196  C
0197  C..... P - FILTER .....
0198  C
0199      WRITE(IW,120)
0200      NA = 1
0201      ITYP = IFIX(D(ISYS,20))
0202      TS   = D(ISYS-2,2)
0203      WRITE(IW,285)ITYP,TS
0204      IF(ITYP.NE. 1) GO TO 60

```



```

0205          ITYPS=IFIX(D(ISYS,21))
0206          GAIN =      D(ISYS,22)
0207          T(1) =      D(ISYS,23)
0208          T(2) =      D(ISYS,24)
0209          T(3) =      D(ISYS,25)
0210          WRITE(IW,290)ITYPS,GAIN,(T(J),J=1,3)
0211          CALL ZTRAN(ITYPS,TS,GAIN,T,A,B,NA,NB)
0212          D(ISYS,30)=FLOAT(NA)
0213          D(ISYS,31)=B(1)
0214          D(ISYS,32)=B(2)
0215          D(ISYS,33)=B(3)
0216          D(ISYS,34)=B(4)
0217          D(ISYS,35)=A(1)
0218          D(ISYS,36)=A(2)
0219          D(ISYS,37)=A(3)
0220          D(ISYS,38)=A(4)
0221      60  WRITE(IW,295)NA,(D(ISYS,J),J=31,34),(D(ISYS,J),J=35,38)
0222  C
0223  C..... R - FILTER.....
0224  C
0225          WRITE(IW,120)
0226          WRITE(IW,300)(D(ISYS,J),J=50,54)
0227          WRITE(IW,301)D(ISYS,60)
0228          WRITE(IW,302)(D(ISYS,J),J=61,63)
0229          WRITE(IW,303)(D(ISYS,J),J=64,66)
0230      65  ISYS=ISYS
0231          WRITE(IW,120)
0232          IW=IR
0233  C.....INPUT OUTPUT SECTION
0234  C
0235      70  WRITE(IW,130)ICHAR,ICHARS,IG
0236          READ (IR,310)ICHAR2
0237  C
0238  C.....CHANGE , CHANGES , RESET , PRINT ,.....
0239  C
0240          IF(ICHAR2 .EQ. KC      ) GO TO 75
0241          IF(ICHAR2 .EQ. KS      ) GO TO 80
0242          IF(ICHAR2 .EQ. KR      ) GO TO 15
0243          GO TO 70
0244  C
0245  C..... SINGLE INPUT
0246  C
0247      75  WRITE(IW,140)
0248          READ(IR,*)N1,AIN(1)
0249          IF(N1.LT.1 .OR. N1 .GT. 100) WRITE(IW,135)
0250          IF(N1.LT.1 .OR. N1 .GT. 100) GO TO 70
0251          D(ISYS,N1)=AIN(1)
0252          GO TO 70
0253  C
0254  C.....MULTI-INPUT 10 NUMBER.....
0255  C

```



```

0256      80  WRITE(IW,145)
0257          READ(IR,*)N1,(AIN(J),J=1,10)
0258          IF(N1.GT.90) WRITE(IW,135)
0259          IF(N1 .GT. 90) GO TO 70
0260          DO 85 J=1,10
0261              JM1=J+N1-1
0262              D(ISYS,JM1)=AIN(J)
0263      85  CONTINUE
0264          GO TO 70
0265  C
0266  C
0267      90  CONTINUE
0268  C      CALL CLOSE(IDCIB,IERR)
0269  C      CALL OPEN(IDCIB,IERR,IFILE,0,77,146,170)
0270          CALL RWNDF(IDCIB,IERR)
0271          IF(IERR .GE. 0) GO TO 91
0272          STOP 4
0273      91  DO 95 J=1,6
0274  C
0275  C
0276          DO 820 I = 1,40
0277              BUF(I) = D(J,I)
0278      820  CONTINUE
0279              CALL WRITF(IDCIB,IERR,BUF,80)
0280          DO 822 I = 41,85
0281              IDUM = I - 40
0282              BUF(IDUM) = D(J,I)
0283      822  CONTINUE
0284              CALL WRITF(IDCIB,IERR,BUF,90)
0285          IF(IERR .GE. 0) GO TO 97
0286          CALL FMGER(IERR)
0287          STOP 4
0288      97  CONTINUE
0289          WRITE(IW,305)J
0290      95  CONTINUE
0291          CALL CLOSE(IDCIB,IERR)
0292          STOP
0293      100  FORMAT(5G12.5)
0294      105  FORMAT(I7,10G12.5)
0295      110  FORMAT(5G12.5)
0296      115  FORMAT(5(G12.5,1X))
0297      120  FORMAT(1X,60(1H-))
0298      125  FORMAT(1X,'MODEL  ,  SYSTEM  ,  CONTROL-ACTION  ,  RUN ',/,
0299          1      1X,'-      -      -      -')
0300      130  FORMAT(1X,'CHANGE  ,  CHANGES  ,  RESET  ,  <',2A1,I1,'> ',/,
0301          &      1X,'-      -      -')
0302      135  FORMAT(1X,'WRONG NUMBER OF INDEX TRY AGIAN')
0303      140  FORMAT(1X,'INPUT INDEX  ,  VALUE')
0304      145  FORMAT(2X,'INPUT INDEX  ,  VALUES .....<><>')
0305      150  FORMAT('<01> TYPE OF SIMULATION =' ,I2)
0306      1500 FORMAT('1',/////,' * * * ',A4,'MODEL * * *',////)

```



```

0307 155 FORMAT(1X,'<10> N( B1 )=',I2,/,
0308 1 (1X,'<11> B1 =',5G12.5) )
0309 160 FORMAT(1X,'<20> N( B2 )=',I2,/,
0310 1 (1X,'<21> B2 =',5G12.5) )
0311 165 FORMAT(1X,'<30> N( D1 )=',I2,/,
0312 1 (1X,'<31> D1 =',5G12.5) )
0313 170 FORMAT(1X,'<40> N( C1 )=',I2,/,
0314 1 (1X,'<41> C1 =',5G12.5) )
0315 175 FORMAT(1X,'<50> N( A )=',I2,/,
0316 1 (1X,'<51> A =',5G12.5) )
0317 180 FORMAT(1X,'<10> T-RK =',F2.0,' FOR G(11)',/,
0318 1 1X,'<11> GAIN =',G12.5,'<12-14> T=',3G12.5)
0319 185 FORMAT(1X,'<20> T-RK =',F2.0,' FOR G(12)',/,
0320 1 1X,'<21> GAIN =',G12.5,'<22-24> T=',3G12.5)
0321 190 FORMAT(1X,'<30> T-RK =',F2.0,' FOR G( D)',/,
0322 1 1X,'<31> GAIN =',G12.5,'<32-34> T=',3G12.5)
0323 195 FORMAT(1X,'<40> T-RK =',F2.0,' FOR G( C)',/,
0324 1 1X,'<41> GAIN =',G12.5,'<42-44> T=',3G12.5)
0325 200 FORMAT(1X,'<02> T-SA =',G12.5,
0326 1 1X,'<03> RK-DT=',G12.5,'<04> RK-ND=',F3.0,/,
0327 2 1X,'<05> ST-DI=',G12.5,' <06> N-LIM=',G12.5)
0328 205 FORMAT(1X,'<07> K11=',F3.0,' <08> K12=',F3.0,
0329 & 1X,'<09> KD =',F3.0,1X,F3.0)
0330 210 FORMAT(' <80> ADC Y=',G12.5,' * SADC + ',G12.5,/,
0331 & ' <82> DAC U.S=',G12.5,' * UN + ',G12.5)
0332 2150 FORMAT('1',/////, ' * * * ',A4,'SYSTEM * * *',///)
0333 215 FORMAT('<01> NG1=',I2,' <02> NF =',I2,' <03> ND =',I2,/,
0334 1 1X,'<04> NH =',I2,' <05> NG2=',I2,/,
0335 2 1X,'<06> K1 =',I2,' <07> K2 =',I2,' <08> KD =',I2,/,
0336 3 1X,' FIX-N =',I2,' <09>NDEL=',I2 )
0337 220 FORMAT(1X,'<10> G1 =',5G12.5)
0338 225 FORMAT(1X,'<20> F =',5G12.5)
0339 230 FORMAT(1X,'<30> D =',5G12.5)
0340 235 FORMAT(1X,'<40> H =',5G12.5)
0341 240 FORMAT(1X,'<50> G2 =',5G12.5)
0342 245 FORMAT(1X,'<60> TYPE OF IDENT=',I2,' ',A4,/,
0343 1 1X,'<61> ROW-1=',G12.5,' <62> ROW-2=',G12.5,/,
0344 2 1X,'<63> P-INT=',G12.5,' <64> I-FIX=',2I3,/,
0345 3 1X,'<66> G*COP=',G12.5,' <67> L-PHI=',2G12.5)
0346 250 FORMAT(1X,'<70> TYPE OF SET-POINT=',F2.0,/,
0347 1 1X,'<71> HIEGHT =',G12.5,' <72> LEINGHT=',G12.5,/,
0348 2 1X,'<73> START =',G12.5,' <74> FINAL =',G12.5,/,
0349 3 1X,'<75> BASE =',G12.5)
0350 255 FORMAT(1X,'<80> TYPE OF DISTURBANCE=',F2.0,/,
0351 1 1X,'<81> HIEGHT =',G12.5,' <82> LEINGHT=',G12.5,/,
0352 2 1X,'<83> START =',G12.5,' <84> FINAL =',G12.5,/,
0353 3 1X,'<85> BASE =',G12.5)
0354 2600 FORMAT('1',/////, ' * * * ',A4,'CONTROL * * *',///)
0355 260 FORMAT('<01> TYPE OF CONTROLLER=',I2,4X,A4,/,
0356 1 1X,'<02> LOWER-U=',G12.5,' <03> UPPER-U=',G12.5,/,
0357 2 1X,'<04> PERCENT=',G12.5,' USTEP =',G12.5)

```



```

0358 265 FORMAT(1X,'<05> TYPE OF PID=',I2,10X,' 0 --> Z-TRAN',/,
0359      1      1X, 28X,      ' 1 --> S-TRAN',/,
0360      2      1X,'<06> PID-ACTION=',G12.5)
0361 270 FORMAT(1X,'<07> KP   =',G12.5,/,
0362      1      1X,'<08> KI   =',G12.5,/,
0363      2      1X,'<09> KD   =',G12.5)
0364 275 FORMAT(1X,'Q-FILTER....',/,1X,'<10> NQ   =',I2,/,
0365      1      1X,'<11> QD(Z)=' ,5G12.5,(/,10X,5G12.5))
0366 280 FORMAT(1X,'<15> QN(Z)=' ,5G12.5,(/,10X,5G12.5))
0367 285 FORMAT(1X,'P-FILTER....',/,
0368      &      1X,'<20> TYPE =',I2,10X,' 0 ---> Z-TRAN',/,
0369      &      1X,'      T-S  =',G12.5 , ' 1 ---> S-TRAN')
0370 290 FORMAT(1X,'<21> T-RK=',I2,/,
0371      &      1X,'<22> GAIN =',G12.5,' <23-25> T=' ,3G12.5)
0372 295 FORMAT(1X,'<30> NP   =',I2,/,
0373      &      1X,'<31> PN   =',4G12.5,/,
0374      &      1X,'<35> PD   =',4G12.5)
0375 300 FORMAT(1X,'R FILTER....',/,1X,'<50> T-RK=' ,F2.0,/,
0376      1 1X,'<51>  GAIN=' ,G12.5,' <52-53> T=' ,2G12.5,'<54> K=' ,F2.0)
0377 301 FORMAT(1X,'<60> U-STEADY-STATE=' ,G12.5)
0378 302 FORMAT(1X,'      PID SETTINGS (AKPS,AKIS,AKDS)',/,
0379      &      1X,'      POSITIONAL OR INCREMENTAL CONTROLLER',/,
0380      1 1X,'<61-63> ',3G12.5)
0381 303 FORMAT(1X,'      Q-FILTER SETTINGS (AKP,AKI,AKD)',/,
0382      &      1X,'<64-66> ',3G12.5)
0383 305 FORMAT(' SECTOR ',I2,' HAS BEEN WRITTEN ')
0384 310 FORMAT(2A1,A2)
0385 315 FORMAT(I1)
0386      END

```


&ERSUM T=00004 IS ON CR00146 USING 00011 BLKS R=0087

```

0001  FTN4
0002      PROGRAM ERSUM( ,101)
0003  C
0004      DIMENSION IDCB(144),BUF(56),ISTR(20),
0005      *          I1(20),I2(20),IR(20),
0006      *          IFILE(3),LU(5),
0007      *          FEED1(20),FEED2(20),SIAE(20)
0008  C
0009      CALL RHPAR(LU)
0010      LUN=LU(1)
0011  C
0012  C
0013      WRITE(LUN,1000)
0014      READ(LUN,1002) (IFILE(I),I=1,3),ICR
0015  C
0016      WRITE(LUN,1006)
0017      READ(LUN,*) MAXREG
0018  C
0019      WRITE(LUN,1010)
0020      READ(LUN,*) ILU
0021
0022  C
0023  C
0024      DO 50 IREG = 1,MAXREG
0025      WRITE(LUN,1012) IREG
0026      READ(LUN,*) I1(IREG),I2(IREG),IR(IREG)
0027  50  CONTINUE
0028  C
0029  C
0030  C
0031  C
0032      CALL OPEN(IDCB,IERR,IFILE,0,77,ICR,144)
0033      IF(IERR.GE.0) GOTO 70
0034      CALL FMGER(IERR)
0035      STOP
0036  C
0037  C-  READ HEADER RECORD.
0038  C
0039  70  CONTINUE
0040      CALL READF(IDCB,IERR,BUF,40,ILEN)
0041      IF(IERR.LT.0) CALL FMGER(IERR)
0042      WRITE(ILU,1020) (BUF(I),I=1,20)
0043      DO 900 IREG = 1,MAXREG
0044  C
0045      IBEG = I1(IREG)
0046      IEND = 0
0047      IDUM = IREG - 1
0048      IF(IREG .NE. 1) IEND = ISTR(IDUM) + IR(IDUM) - 1

```



```

0049      IRW = IBEG - IEND -1
0050      CALL POSNT(IDCIB,IERR,IRW,0)
0051          CALL HSIAE(IDCIB,BUF,ISTR(IREG),BEG,END,
0052      *          SIAE(IREG),FEED1(IREG),FEED2(IREG),
0053      *          I1(IREG),I2(IREG),IR(IREG))
0054      WRITE(ILU,1022) IREG,BEG,END
0055 1022  FORMAT(' REGION ',I4,' BEGINS AT ',F7.2,' ENDS AT ',F7.2)
0056  C
0057  C
0058  900  CONTINUE
0059  C
0060  C-  PRINT OUT SIAE RESULTS.
0061  C
0062      CALL RWNDF(IDCIB,IERR)
0063      CALL READF(IDCIB,IERR,BUF,40,ILEN)
0064      WRITE(ILU,951) (BUF(I),I=1,20)
0065  C
0066          DO 950  IREG = 1,MAXREG
0067              WRITE(ILU,953) IREG,I1(IREG),I2(IREG),
0068      *          ISTR(IREG),FEED1(IREG),FEED2(IREG),
0069      *          IR(IREG),SIAE(IREG)
0070  950      CONTINUE
0071  C
0072      CALL CLOSE(IDCIB,IERR)
0073      STOP
0074  C
0075  C-----
0076  C-
0077  C-  FORMAT STATEMENTS.
0078  C-
0079  C
0080  951  FORMAT(//,' * * FOR FILE ON * * ',/,20A4,/)
0081  953  FORMAT(/,' FOR REGION ',I2,5X,' FROM: ',I4,' TO: ',I4,
0082      *      /,' FEED CHANGE AT ',I4,5X,' FROM: ',F6.3,' TO: ',F6.3
0083      *      /,' SIAE AFTER ',I2,' ITERATION IS =',F10.4)
0084  1000 FORMAT(/,' ENTER FILENAME(3A2) AND CARTRIDGE(I3): ', '_')
0085  1002 FORMAT(3A2,I3)
0086  1006 FORMAT(/,' ENTER NO. OF REGION (MAX=20): ', '_')
0087  1010 FORMAT(/,' ENTER OUTPUT DEVICE: ', '_')
0088  1012 FORMAT(/,' FOR REGION: ',I2,
0089      *      /,' ENTER START,FINAL AND NO. OF ITERATIONS: ', '_')
0090  1020 FORMAT(/,' PLOTTING FOR ',/,20A4)
0091      END

```


&HPL01 T=00004 IS ON CR00136 USING 00028 BLKS R=0000

```

0001 C
0002 C* THIS PROGRAM PLOTS DATA STORED IN DATA FILE.
0003 C
0004 FTN4
0005     PROGRAM HPL01( ,101)
0006 C
0007     DIMENSION IDC(144),BUF(56),X(702),Y(702),
0008     *           I1(15),I2(15),IR(15),
0009     *           SPAN(6),ZERO(6),IFLO(6),Y00(6),
0010     *           IPOS(11),SSTART(11,15),RRANGE(11,15),
0011     *           NPAR(5),IFILE(3),LU(5),
0012     *           DSTART(6),DRANGE(6)
0013 C
0014     DATA IPOS/9,10,12,11,13,14,17,0,0,0,0/,
0015     *      CONA/1.0/,CONB/0./,
0016     *      IFLO/0,1,1,0,1,1/,
0017     *      SPAN/.12857,2.2975,2.5517,1.,2.50,1.3140/,
0018     *      ZERO/89.167,0.,0.,0.,0.,0./,
0019     *      SSTART/110*0./,RRANGE/110*0./,
0020     *      Y00/0.,4.,5.5,0.,4.,5.5/,
0021     *      DSTART/93.5,0.00,12.0,1.00,0.00,0.00/,
0022     *      DRANGE/1.00,0.00,12.00,3.00,0.00,0.00/
0023     CALL RMPAR(LU)
0024     LUN=LU(1)
0025 C
0026 C* OBTAIN INFORMATION ON WHERE DATA IS STORED
0027 C
0028     WRITE(LUN,1000)
0029     READ(LUN,1002) (IFILE(I),I=1,3),ICR
0030 C
0031     WRITE(LUN,1004)
0032     READ(LUN,*) (NPAR(I),I=1,5)
0033 C
0034     WRITE(LUN,1006)
0035     READ(LUN,*) MAXREG
0036 C
0037     WRITE(LUN,1008)
0038     READ(LUN,*) IYOPT
0039 C
0040     WRITE(LUN,1010)
0041     READ(LUN,*) ILU
0042 C
0043     WRITE(LUN,1011)
0044     READ(LUN,*) ST
0045
0046 C
0047 C* READ IN INFORMATION FOR EACH REGION.
0048 C

```



```

0049          DO 50 IREG = 1,MAXREG
0050          WRITE(LUN,1012) IREG
0051          READ(LUN,*) I1(IREG),I2(IREG),IR(IREG)
0052 C
0053          WRITE(LUN,1014)
0054          READ(LUN,*)IDUM
0055          IF(IDUM.NE.1) GO TO 20
0056 C
0057 C*  OBTAIN RANGE AND START PONT OF EACH REGION FOR PROCESS MEASURE
0058 C
0059          WRITE(LUN,1016) MAXREG
0060          READ(LUN,*) (SSTART(I,IREG),I=1,6)
0061          WRITE(LUN,1017) MAXREG
0062          READ(LUN,*) (RRANGE(I,IREG),I=1,6)
0063 C
0064      20      IF(IDUM.NE.2) GO TO 22
0065 C
0066 C- TAKE DEFAULT SCALING.
0067 C
0068          DO 24 I= 1,6
0069          SSTART(I,IREG) = DSTART(I)
0070          RRANGE(I,IREG) = DRANGE(I)
0071      24      CONTINUE
0072 C
0073      22      WRITE(LUN,1018)
0074          READ(LUN,*) IDUM
0075          IF(IDUM.NE.1) GO TO 30
0076 C
0077 C*  WANT START AND RANGE OF PARAMETERS.
0078 C
0079          WRITE(LUN,1016) MAXREG
0080          READ(LUN,*) (SSTART(I,IREG),I=7,11)
0081          WRITE(LUN,1017) MAXREG
0082          READ(LUN,*) (RRANGE(I,IREG),I=7,11)
0083 C
0084      30      CONTINUE
0085 C
0086      50      CONTINUE
0087 C
0088 C*  OBTAIN THE STARTING POSITION OF EACH TYPE OF PARAMETER
0089 C
0090          DO 60 I = 8,11
0091          IDUM = I - 1
0092          IDU1 = I - 7
0093          IPOS(I) = IPOS(IDUM) + NPAR(IDU1)
0094      60      CONTINUE
0095 C
0096 C
0097          CALL OPEN(IDCIB,IERR,IFILE,0,77,ICR,144)
0098          IF(IERR.GE.0) GOTO 70
0099          CALL FMGER(LUN,IERR)

```



```

0100      STOP
0101      C
0102      C-  READ HEADER RECORD.
0103      C
0104      70  CONTINUE
0105          CALL READF(IDCB,IERR,BUF,40,ILEN)
0106          IF(IERR.LT.0) CALL FMGER(LUN,IERR)
0107          WRITE(LUN,1020) (BUF(I),I=1,20)
0108          DO 900  IREG = 1,MAXREG
0109      C
0110          WRITE(LUN,2040)
0111          IBEG = I1(IREG)
0112          IEND = 0
0113          IF(IREG .NE. 1) IEND = I2(IREG-1)
0114          IRW = IBEG - IEND - 2
0115          CALL POSNT(IDCB,IERR,IRW,0)
0116          CALL LOCF(IDCB,IERR,IREG,IRB,IOFF)
0117      C
0118      C*  FILL IN TIME INFORMATION (X-AXIS).
0119      C
0120          J = 0
0121          DO 100 I = I1(IREG),I2(IREG),IR(IREG)
0122              J = J + 1
0123              X(J) = FLOAT(I-I1(IREG))*ST
0124      100  CONTINUE
0125      C
0126      C*  COMPUTE NO. OF POINTS TO BE PLOTTED
0127      C
0128          NPT = I2(IREG) - I1(IREG) + 1
0129      C
0130      C*  INITIALIZE PLOTTING.
0131      C
0132          CALL PLOTS(0.0,ILU)
0133          IF(ILU.EQ.4) CALL FACTOR(.6)
0134          CALL PLOT(1.87,1.61,-3)
0135          CALL HCALE(X,5.0,NPT,1)
0136      C
0137      C*  DETERMINE WHICH OPTION TO PLOT.
0138      C
0139          IBEG = 1
0140          IEND = 3
0141          IF(IYOPT.EQ.1 .OR. IYOPT.EQ.3) GOTO 150
0142      C
0143          IBEG = 4
0144          IEND = 6
0145          IF(IYOPT.EQ.2) GOTO 150
0146      C
0147          IBEG = 1
0148      150  CONTINUE
0149      C
0150      C

```



```

0151 C-  START PLOTTING THE FLOW MEASUREMENT.
0152 C
0153          DO 200 I = IBEG,IEND
0154          ITYP = I
0155 C
0156 C-  IF ONLY PLOTTING BOTTOM, PLOT FEED INSTEAD OF REFLUX.
0157 C
0158          IF(IYOPT.EQ.2 .AND. ITYP.EQ.6) ITYP = 3
0159          IF(IYOPT.EQ.3 .AND. ITYP.EQ.2) ITYP = 4
0160          START = SSTART(ITYP,IREG)
0161          RANGE = RRANGE(ITYP,IREG)
0162 C
0163 C*  CALL SUBROUTINE TO PLOT OUT MEASUREMENT FOR EACH REGION.
0164 C
0165          CALL HPLT(IDCIB,BUF,X,Y,IREF,IRB,IOFF,
0166          *          I1(IREF),I2(IREF),IR(IREF),BEG,END,
0167          *          IPOS(ITYP),SPAN(ITYP),ZERO(ITYP),Y00(I),
0168          *          IFLO(ITYP),START,RANGE,ITYP)
0169          IF(ITYP.EQ.3.AND.IEND.EQ.6.AND.IYOPT.EQ.4) CALL PLOT(0.,-9.
0170          WRITE(LUN,2000) ITYP,IREF
0171          WRITE(LUN,2020) BEG,END
0172 2020          FORMAT(' PLOTTING FROM ',F7.2,' TO ',F7.2)
0173 200      CONTINUE
0174 C
0175 C*  PLOT OUT PARAMETERS FOR EACH REGION.
0176 C
0177          IF(ILU .EQ. 4) GO TO 800
0178          CALL PLOT(0.,-9.5,-3)
0179          CALL PLOT(1.397,.2733,-3)
0180 C
0181 C
0182          DO 300 I = 1,5
0183          ITYP = 6 + I
0184 C
0185          START = SSTART(ITYP,IREF)
0186          RANGE = RRANGE(ITYP,IREF)
0187 C
0188 C*  CALL SUBROUTINE TO PLOT OUT PARAMETERS FOR EACH REGION.
0189 C
0190          CALL HPLT(IDCIB,BUF,X,Y,IREF,IRB,IOFF,
0191          *          I1(IREF),I2(IREF),IR(IREF),BEG,END,
0192          *          IPOS(ITYP),CONA,CONB,Y00(1),NPAR(I),0,
0193          *          START,RANGE,ITYP)
0194          WRITE(LUN,2000)ITYP,IREF
0195          WRITE(LUN,2020)BEG,END
0196 C
0197 300      CONTINUE
0198          CALL PLOT(-1.397,-.2733,-3)
0199 800      CALL PLOT(0.,0.,999)
0200 C
0201 C

```



```

0202  900  CONTINUE
0203  C
0204      CALL CLOSE(IDCIB,IERR)
0205      STOP
0206  C
0207  C-----
0208  C-
0209  C-  FORMAT STATEMENTS.
0210  C-
0211  C
0212  1000 FORMAT(/,' ENTER FILENAME(3A2) AND CARTRIDGE(I3): ','_')
0213  1002 FORMAT(3A2,I3)
0214  1004 FORMAT(/,' ENTER NO. OF G1,F,D,H,G2 PARAMENTERS: _')
0215  1006 FORMAT(/,' ENTER NO. OF REGION (MAX=10): ','_')
0216  1008 FORMAT(/,' ENTER 1 = TOP, 2 = BOTTOM, 3 = FD,TC,BC, 4 = ALL:
0217  1010 FORMAT(/,' ENTER PLOT DEVICE (4=HP2648 6=LP): ','_')
0218  1011 FORMAT(/,' ENTER SAMPLE TIME IN MINUTES:_')
0219  1012 FORMAT(/,' FOR REGION: ',I2,
0220      *,'/',' ENTER START, FINAL AND REPEAT FACTOR: ','_')
0221  1014 FORMAT(/,' ENTER: 0-AUTOSCALE 1-MANUAL ENTRY 2-DEFAULT: ','')
0222  1016 FORMAT(' ENTER ',I2,' START VALUE FOR REGIONS, 0 = AUTO SCAL
0223  1017 FORMAT(' ENTER ',I2,' RANGE FOR DIFFERENT REGION (UNITS/TICK
0224  1018 FORMAT(/,' WANT TO ENTER SCALE FACTOR FOR PARAM.',
0225      *      '(1 - YES): _')
0226  1020 FORMAT(' * * PLOTTING FOR * *',/,20A4)
0227  2000 FORMAT(/,' * * COMPLETED TYPE ',I2,' FOR REGION ',I2,' * *')
0228  2040 FORMAT(/)
0229      END

```


&HPLT2 T=00004 IS ON CR00136 USING 00017 BLKS R=0000

```

0001  C
0002  C*  THIS SUBROUTINE DOES THE ACTUAL PLOTTING FOR EACH REGION.
0003  C
0004  FTN4
0005      SUBROUTINE HPLT(IDCB,BUF,X,Y,IREC,IRB,IOFF,
0006      *                I1,I2,IR,BEG,END,
0007      *                IPOS,SPAN,ZERO,Y00,NPAR,IFLO,
0008      *                START,RANGE,ITYP)
0009  C
0010      DIMENSION IDCB(1),BUF(1),X(1),Y(1)
0011      DATA ILEN/112/,IFLG/1/
0012      IF(NPAR .EQ. 0) GO TO 500
0013  C
0014  C
0015      NPT = I2 - I1 + 1
0016      K = 0
0017  C
0018  C
0019      DO 100  II = 1,NPAR
0020          J = 0
0021          CALL APOSN(IDCB,IER,IRCD,IRB,IOFF)
0022          CALL READF(IDCB,IERR,BUF,20,ILEN)
0023          IVAL = IPOS + K
0024          K = K + 1
0025  C
0026  C
0027          DO 110  I = 1,NPT
0028              CALL READF(IDCB,IERR,BUF,112,ILEN,LEN)
0029              IF(ILEN .EQ. -1) GO TO 900
0030              IF(I .EQ. 1) BEG = BUF(3)
0031              IF(I .EQ. NPT) END = BUF(3)
0032  C
0033              J = J + 1
0034              Y(J) = BUF(IVAL)
0035              IF(IFLO.EQ.1) Y(J) = SQRT(Y(J))
0036              Y(J) = SPAN * Y(J) + ZERO
0037      110      CONTINUE
0038  C
0039  C
0040      IF(K .NE. 4) GO TO 120
0041      CALL PLOT(0.,0.,10)
0042      CALL PLOT(0.,-7.,-3)
0043  C
0044      120  CONTINUE
0045      X0=0
0046      IF(Y00 .EQ. 0.) GO TO 200
0047      Y0 = Y00
0048      GOTO 210

```



```

0049 C
0050 200 Y0 = 3.5
0051 IF(II.EQ.1 .OR. II.EQ.4) Y0 = 0.
0052 C
0053 210 CALL PLOT(X0,Y0,-3)
0054 C
0055 C- START PLOTTING.
0056 C
0057 N=NPT
0058 CALL SYMBOL(1.88,-.55,.15,10HTIME (MIN),0.0,10)
0059 CALL AXIS(0.,0.,20H
0060 * -20,5.0,0.0,X(N+1),X(N+2))
0061 Y(N+1) = START
0062 Y(N+2) = RANGE
0063 IF(START .EQ. 0.) CALL HCALE(Y,3.0,NPT,1)
0064 GOTO(1,2,3,4,5,6,7,8,9,10,11) ITYP
0065 C
0066 1 CALL SYMBOL(-.4,.975,.15,10HXD - [WTZ],90.0,10)
0067 CALL AXIS(0.,0.,0,0,3.0,90.0,Y(N+1),Y(N+2))
0068 GOTO 250
0069 C
0070 2 CALL SYMBOL(-.4,.975,.15,10HRE - [G/S],90.0,10)
0071 CALL AXIS(0.,0.,0,0,3.0,90.,Y(N+1),Y(N+2))
0072 GOTO 250
0073 C
0074 3 CALL SYMBOL(-.4,.975,.15,10HFE - [G/S],90.0,10)
0075 CALL AXIS(0.,0.,0,0,2.0,90.,Y(N+1),Y(N+2))
0076 GOTO 250
0077 C
0078 4 CALL SYMBOL(-.4,.975,.15,10HXB - [WTZ],90.0,10)
0079 CALL AXIS(0.,0.,0,0,3.0,90.,Y(N+1),Y(N+2))
0080 GOTO 250
0081 C
0082 5 CALL SYMBOL(-.4,.975,.15,10HST - [G/S],90.0,10)
0083 CALL AXIS(0.,0.,0,0,3.0,90.,Y(N+1),Y(N+2))
0084 GOTO 250
0085 C
0086 6 CALL SYMBOL(-.4,.975,.15,10HDI - [G/S],90.0,10)
0087 CALL AXIS(0.,0.,0,0,3.0,90.0,Y(N+1),Y(N+2))
0088 GOTO 250
0089 C
0090 7 CALL SYMBOL(-.4,.975,.15,10HPARAM - G1,90.0,10)
0091 CALL AXIS(0.,0.,18H ,18,3.0,90.,
0092 * Y(N+1),Y(N+2))
0093 GOTO 250
0094 C
0095 8 CALL SYMBOL(-.4,.975,.15,10HPARAM - F ,90.0,10)
0096 CALL AXIS(0.,0.,18H ,18,3.0,90.,
0097 * Y(N+1),Y(N+2))
0098 GOTO 250
0099 C

```



```

0100      9  CALL SYMBOL(-.4,.975,.15,10HPARAM - D ,90.0,10)
0101      CALL AXIS(0.,0.,18H ,18,3.0,90.,
0102      *      Y(N+1),Y(N+2))
0103      GOTO 250
0104  C
0105      10  CALL SYMBOL(-.4,.975,.15,10HPARAM - H ,90.0,10)
0106      CALL AXIS(0.,0.,18H ,18,3.0,90.,
0107      *      Y(N+1),Y(N+2))
0108      GOTO 250
0109  C
0110      11  CALL SYMBOL(-.4,.975,.15,10HPARAM - G2,90.0,10)
0111      CALL AXIS(0.,0.,18H ,18,3.0,90.,
0112      *      Y(N+1),Y(N+2))
0113  C
0114      250  CONTINUE
0115      CALL LINE(X,Y,NPT,1,0,1)
0116      100  CONTINUE
0117  C
0118  C
0119      IF(ITYP.EQ.3 .OR. ITYP.GE.6) CALL PLOT(0.,0.,10)
0120  C
0121  C- CHECK IF WE HAVE TO RESET THE ORIGIN.
0122  C
0123      IF(NPAR.EQ.1 .OR. NPAR.EQ.4) Y0 = 0
0124      IF(NPAR.EQ.2 .OR. NPAR.EQ.5) Y0 = -3.50
0125      IF(NPAR.EQ.3 .OR. NPAR.EQ.6) Y0 = -7.
0126      CALL PLOT(0.,Y0,-3)
0127  C
0128      500  CONTINUE
0129      RETURN
0130  C
0131  C
0132      900  WRITE(LUN,999)
0133      999  FORMAT(' REACHED END OF FILE ')
0134      CALL CLOSE(IDCB,IERR)
0135      STOP
0136      END

```


&GCLN2 T=00003 IS ON CR00139 USING 00012 BLKS R=0000

```

0001  FTN4
0002      PROGRAM GCLNK (3,70), GC DATA TRANSFER, 790301, VFB
0003  C
0004  C
0005  C      THIS PROGRAM IS USED TO RECEIVE GC DATA FROM A REMOTE
0006  C      CPU. IT USES CLASS I/O TO RECEIVE THE DATA.
0007  C      THE PROGRAM IS ALWAYS READY TO RECEIVE THE DATA.
0008  C      THE PROGRAM STORES DATA IN REAL TIME COMMON AREA.
0009  C      START OF THE AREA MUST BE SET UP IN REAL TIME
0010  C      COMMON POINTER AREA. CURRENTLY POINTER 4, WORD 8
0011  C      OF RT COMMON IS USED TO POINT TO THE GC DATA AREA.
0012  C      THE LENGTH OF THE AREA IS 80 WORDS.
0013  C
0014  C
0015  C
0016  C
0017      COMMON / / ICOM(1)
0018      DIMENSION IBUF(50,10), ICLAS(10)
0019  C
0020  C      INITIALIZE I/O, CHECK STATUS, LOCK RECEIVING UNIT,
0021  C      START CLASS I/O AND
0022  C      SET UP COMMON ADDRESSING
0023  C
0024      DATA LUGC/33/, NBUF/10/, NEXT/0/
0025      DATA IRLN/50/,LEN/50/
0026      DATA ICLAS/10*100000B/
0027      IAS=ICOM(8)
0028      IAD=IAS+6
0029      ICOM(IAS)=0
0030      LU=LOGLU(ISESN)
0031      CALL EXEC (13,LUGC,IST1,IST2,IST3)
0032      IF (IST3.LT.0) GO TO 800
0033      CALL LURQ(1,LUGC,1)
0034  C      CALL EXEC (22,1)
0035          DO 100 I=1,NBUF
0036              II=I
0037              CALL EXEC(17,LUGC,IBUF(1,I) ,IRLN,IP1,IP2,ICLAS(I))
0038              CALL ABREG (IA,IB)
0039  D      WRITE (LU,1030) IA,IB
0040  D1030      FORMAT (" ALOCATE 1: IA,IB=",208)
0041              IF (IA.LT.0) GO TO 110
0042              ICLAS(I)=IAND(ICLAS(I),077777B)
0043      100      CONTINUE
0044  C
0045      110 NBUF=II-1
0046              WRITE(LU,1035) NBUF
0047      1035      FORMAT(1X,"NUMBER OF BUFFERS=",I3)
0048  C      ICLAS=IOR (ICLAS,20000B)

```



```

0049 C      ICLAS=IAND(ICLAS,077777B)
0050 C
0051 C
0052 C      START OF INPUT LOOP
0053 C
0054 C
0055     500 CONTINUE
0056 C
0057     DO 200 IBUFN=1,NBUF
0058     CALL EXEC(21,ICLAS(IBUFN),IBUF(1,IBUFN),LEN,IP1,IP2,IP3)
0059     CALL ABREG(IA,IBUFL)
0060 D      WRITE (LU,1050) IA,IBUFL
0061 D1050 FORMAT(" GET 1: IA, IBUFL=",208)
0062     ICLAS(IBUFN)=0
0063     IF (IBUFL.EQ.0) GO TO 130
0064     IF (IBUFL.LT.0.OR.IBUFL.GT.40) GO TO 150
0065 D      WRITE (LU,1060) (IBUF(III,IBUFN),III=1,IBUFL)
0066 D1060 FORMAT(1X,20A2)
0067     IF (IAD+IBUFL.GT.IAS+ICOM(9)-1) IAD=IAS+6
0068 D      WRITE(LU,1070) LUGC,IRLEN,LEN,IBUFN,IAD
0069     1070 FORMAT("LUGC,IRLEN,LEN,IBUFN,IAD=" /
0070     25I10)
0071 C
0072     DO 120 J=1,IBUFL
0073     ICOM(IAD+J)=IBUF(J,IBUFN)
0074     120 CONTINUE
0075 C
0076     130 IF (ICOM(IAS).EQ.0) GO TO 140
0077     NEXT=NEXT+1
0078     IF (NEXT.GE.NBUF) GO TO 999
0079     GO TO 200
0080 C
0081     140 IAD=IAD+IBUFL
0082     150 CALL EXEC (17,LUGC,IBUF(1,IBUFN),IRLEN,IP1,IP2,ICLAS(IBUFN))
0083     CALL ABREG (IA,IB)
0084 D      WRITE (LU,1030) IA,IB
0085     200 CONTINUE
0086     GO TO 500
0087 C
0088 C      END OF INPUT LOOP
0089 C
0090 C
0091     800 CALL ABORT (0,1,25H***GC LINK UNIT DOWN ***)
0092     999 CALL LURQ(0,LUGC,1)
0093     STOP
0094     END

```



```

0001  FTN4
0002      PROGRAM GCSCH(3,80), PROGRAM TO INITIATE GC ON COLUMN HYL790
0003      COMMON ICOM(1)
0004      DIMENSION INAME1(3),INAME2(3)
0005      DATA INAME1/2HGC,2HSH,2HT /,INAME2/2HSE,2HGS,2HT /
0006      ISTR=ICOM(8)
0007      ILEN=ICOM(9)-7
0008      LUN=ICOM(ISTR+3)
0009      IMAX=ICOM(ISTR+4)
0010      IDFLT = ICOM(ISTR + 5)
0011  C
0012      CALL BOTCH(ISTR,ILEN,LUN,IMAX)
0013      CALL DOL(1,1,4,4,IERR)
0014      IF(IERR.NE.1) GO TO 500
0015  C
0016  C*  SCHEDULE GCSWT TO RESET GC START BUTTON.
0017  C
0018      CALL EXEC(10+100000B,INAME1)
0019      GO TO 950
0020      STOP
0021  C
0022  C*  IF ERROR DETECTED, TURN PROGRAM OFF AND SET COMPOSITION TO DEFA
0023  C
0024      500  WRITE(LUN,600)IERR
0025      600  FORMAT("PROGRAM GCSCH FAILED TO CLOSE DIGITAL SWITCH, IERR -
0026      GO TO 999
0027  C
0028      950  WRITE(LUN,952)
0029      952  FORMAT("SCHEDULE OF GCSWT UNSUCCESSFUL")
0030  C
0031      999  ICOM(ISTR+1) = IDFLT
0032      CALL EXEC(10,INAME2)
0033      STOP
0034      END
0035  C
0036  C
0037  C
0038      SUBROUTINE BOTCH(ISTR,ILEN,LUN,IMAX)
0039      COMMON ICOM(1)
0040      DIMENSION CON(6), IVAL(4), ITIME(5)
0041      DATA CON/100.,1.,.1,.001,10.,.01/,IBLANK/2H /
0042      ICHR=2H &
0043  C
0044  C*  SEARCH FOR CHARACTER STRING "&".  THE COMPOSITION OF WATER WIL
0045  C*  PRECEDE THIS STRING.
0046  C
0047      IADD=ISTR+6
0048      DO 10 I=1,ILEN
0049      IPRE=I
0050      IF(ICOM(IADD+IPRE).EQ.ICHR) GO TO 12

```



```

0051 10 CONTINUE
0052 GO TO 500
0053 C
0054 C* STORE ASCII REPRESENTATION OF COMPOSITION INTO ARRAY IVAL.
0055 C
0056 12 DO 20 I=1,4
0057     IPOS=IPRE-I
0058     IF(IPOS.LT.0) IPOS=ILEN+IPOS+1
0059     IVAL(5-I)=ICOM(IADD+IPOS)
0060 20 CONTINUE
0061 C
0062 C* CONVERT THE VALUE IN ASCII INTO A DECIMAL NUMBER.
0063 C
0064     VALU=0.
0065     DO 30 I=1,4
0066     IDUM=IAND(ISHFT(IVAL(I),-8),377B)
0067     IF(IDUM.EQ.32) CON(I)=0.
0068     VALU=VALU+CON(I)*FLOAT(IDUM-48)
0069     IF((I.EQ.2).OR.(I.EQ.4)) GO TO 30
0070     IDUM=(I-1)/2+5
0071     IDU1=IAND(IVAL(I),377B)
0072     IF(IDU1.EQ.32) CON(IDUM)=0.
0073     VALU=VALU+CON(IDUM)*FLOAT(IDU1-48)
0074 30 CONTINUE
0075 C
0076 C
0077 C* CONVERT THE COMPOSITION TO AN INTEGER BETWEEN 0-32767 FOR METH
0078 C* NOTE: 1% = 1000.
0079 C
0080     VALU=100.-VALU
0081     IMET=IFIX(VALU*1000.)
0082     IF (LUN.NE.34) WRITE(LUN,103) VALU
0083 103 FORMAT(10X,"XB = ",F6.3)
0084 C
0085 C* CHECK IF THE VALUE HAS CHANGED.
0086 C
0087     IF(IMET.EQ.ICOM(ISTR+1)) GO TO 200
0088     IF(IMET .LT. 0) GO TO 220
0089 C
0090 C- CHECK THAT GC READING HAS NOT CHANGE BY MAX MAX IS STORED IN
0091 C- WORD OF GC AREA.
0092 C
0093     IDIFF = IABS(ICOM(ISTR + 1) - IMET)
0094     IF(IDIFF. GT. ICOM(ISTR + 6) ) GOTO 300
0095 C
0096     ICOM(ISTR + 2) = 0
0097     ICOM(ISTR+1)=IMET
0098 C
0099 C- BLANK OUT GC AREA IN RT COMMON.
0100 C
0101     DO 25 I=1,ILEN

```



```

0102      II = I
0103      ICOM(IADD + II) = IBLANK
0104      25  CONTINUE
0105      C
0106      C
0107      RETURN
0108      C
0109      C*  IF GC READING REMAINS THE SAME FOR IMAX TIME, GC COMPUTER IS P
0110      C*  DOWN, PUT OUTPUT SEGMENT OF STEAM FLOW ON MANUAL AND GIVE MESS
0111      C
0112      200  ICOM(ISTR+2)=ICOM(ISTR+2)+1
0113           IF(ICOM(ISTR+2).LT.IMAX) RETURN
0114           WRITE(LUN,205)ICOM(ISTR+2)
0115           ICOM(ISTR+1) = ICOM(ISTR+5)
0116           RETURN
0117      C
0118      C
0119      220  WRITE(LUN,222)
0120           GOTO 600
0121      C
0122      C
0123      300  WRITE(LUN,310)
0124           GOTO 600
0125      C
0126      C*  IF I EQUAL TO ILEN, SEARCH FOR WATER COMPOSITION FAILED.
0127      C
0128      500  CALL EXEC(11,ITIME)
0129           IT=ITIME(4)*100+ITIME(3)
0130           WRITE(LUN,104)IT
0131      104  FORMAT("CANNOT FIND BOTTOM COMPOSITOM OF WATER AT",I5)
0132      C
0133      C
0134      600  IBEG = IADD + 1
0135           IEND = IADD + ILEN
0136           WRITE(LUN,612) (ICOM(I) , I = IBEG,IEND)
0137           GO TO 200
0138      C
0139      C FORMAT STATEMENTS.
0140      C
0141      205  FORMAT(' READING FROM GC HAS NOT CHANGE FOR',I5,/,
0142      *      ' COMPOSITION SET TO DEFAULT.')
0143      222  FORMAT(' GC OUTPUT IS NEGATIVE')
0144      310  FORMAT(' GC READING CHANGE GREATER THAN MAX ')
0145      612  FORMAT(40A2,/,40A2)
0146      END

```



```
0001  FTN4
0002      PROGRAM GCSWT(3,80), GC AUTO. SAMPLING INITIATION PROGRAM  D
0003      CALL WAIT(5,2,IERR)
0004      CALL DOL(1,1,0,4,IERR)
0005      IF(IERR.NE.1) WRITE(1,100) IERR
0006      STOP
0007  100  FORMAT(' GCSWT CALL DOL ERROR, IERR = ',I2)
0008      END
```



```

0001  FTN4
0002      PROGRAM BDC99(3,85), DISTILLATION COLUMN MONITOR PROGRAM  HK
0003  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004  C
0005  C          THIS PROGRAM MONITORS THE DISTILLATION COLUMN IN THE
0006  C          ABSENCE OF AN OPERATOR. THE KEY VARIABLES ARE:
0007  C
0008  C          KEY VARRIABLES   ICHAN NO.   I.D.   HILIM   LOLIM   IDWN2
0009  C          FEED FLOW LOOP      12       1     4900    1000     6MIN
0010  C          DISTILLATE FLOW LOOP 10       2     4900    1000    30MIN
0011  C          BOTTOM FLOW LOOP     8       3     5100    1000    30MIN
0012  C          REFLUX FLOW LOOP    13      4     5000    1000    30MIN
0013  C          STEAM FLOW LOOP      9       5     3800    1200    15MIN
0014  C          COOLING WATER LOOP  11      6     5100    1200    15MIN
0015  C          TOWER PRESSURE LOOP  16      7     4900    1200    15MIN
0016  C          REBOILER LEVEL LOOP 17      8     4900    1100    15MIN
0017  C          CONDENSER LEVEL LOOP 18      9     4900    1100    30MIN
0018  C
0019  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0020      COMMON ICOM(1)
0021      REAL DCDATA(11),CONA(11),CONB(11)
0022      INTEGER IPARAM(5),ITIME(5),IDATE(15),ITY1(11),ITY2(11),
0023  $ ICHAN(11),IDATA(11),IDWN1(9),IDWN2(9),HILIM(9),LOLIM(9)
0024      DATA ICHAN/12,10,8,13,9,11,16,17,18,15,0/
0025      DATA CONA/2.5777,1.5205,1.5105,2.2975,2.25,71.278,
0026  $ 0.96988,1.0212,1.0723,0.12857,0.0010/
0027      DATA CONB/9*0.0,89.167,0.0/
0028      DATA NDATA,NTEST/11,9/,ITY1/10*1,2/,ITY2/8*1,3*2/
0029      DATA IDWN1/9*0/,IDWN2/5,3*30,4*15,30/,IDWN/0/
0030      DATA HILIM/2*4900,5100,3800,5000,5100,3*4900/
0031      DATA LOLIM/4*1000,3*1200,2*1100/
0032  C
0033  C          FORMAT STATEMENTS
0034  C
0035  2000  FORMAT(/,5X,"BDC99 UP @ ",15A2)
0036  2010  FORMAT(/,1X,"TIME",2X,"C.L. ERR",1X," 1 - FE ",1X," 2 - TP "
0037  $ " 3 - BP ",1X," 4 - RE ",1X," 5 - ST ",1X," 6 - CW ",1X,
0038  $ " 7 - PR ",1X," 8 - RL ",1X," 9 - CL "1X,"10 - XD ",1X,
0039  $ "11 - XB ")
0040  2020  FORMAT(1X,2I2,12(1X,F8.3))
0041  2030  FORMAT(///)
0042  C
0043  C          INITIALIZATION
0044  C
0045      CALL RMPAR(IPARAM)
0046      LUN=IPARAM(1)
0047      MULT=IPARAM(2)
0048      IRES=3
0049      ICHAN(11)=ICOM(8)+1
0050      DO 5000 I=1,9
0051      IDWN2(I)=IDWN2(I)/MULT

```



```

0052 5000 CONTINUE
0053      CALL FTIME(IDATE)
0054      WRITE(1,2000) IDATE
0055      WRITE(LUN,2000) IDATE
0056      WRITE(LUN,2010)
0057      ISKIP=0
0058 C
0059 C      DATA ACQUISITION
0060 C
0061 5010 CALL ACQUI(LUN,NDATA,ITY1,ITY2,ICHAN,IDATA,DCDATA,CONA,CONB,
0062      $IERR)
0063      IF (IERR.NE.1) GO TO 5005
0064      CALL EXEC(11,ITIME)
0065      ERR=100.0*(DCDATA(1)-DCDATA(2)-DCDATA(3))/DCDATA(1)
0066      WRITE(LUN,2020) ITIME(4),ITIME(3),ERR,(DCDATA(I), I=1,NDATA)
0067 C
0068 C      VARIABLE LIMIT CHECKING
0069 C
0070      CALL LMTCK(LUN,NTEST,IDATA,HILIM,LOLIM,IDWN1,IDWN2,IDOWN,
0071      $ISKIP)
0072      IF(IDOWN.NE.0) GO TO 5005
0073 C
0074 C      SUSPEND THE PROGRAM FOR MULT-MIN.
0075 C
0076      CALL WAIT(MULT,IRES,IERR)
0077      IF (IERR.NE.1) GO TO 5005
0078      ISKIP=ISKIP+1
0079      IF (ISKIP.NE.45) GO TO 5010
0080      ISKIP=0
0081      WRITE(LUN,2030)
0082      WRITE(LUN,2010)
0083      GO TO 5010
0084 C
0085 C      DISTILLATION COLUMN SHUT DOWN
0086 C
0087 5005 CONTINUE
0088      CALL DOL(1,1,8,8,IERR)
0089 C
0090 C      RESET THE ECO 15 SEC. AFTER THE SHUT DOWN
0091 C
0092      CALL WAIT(15,2,IERR)
0093      CALL DOL(1,1,0,8,IERR)
0094      STOP
0095      END
0096 C
0097 C
0098      SUBROUTINE LMTCK(LUN,NDATA,IDATA,HILIM,LOLIM,IDWN1,IDWN2,IDO
0099      $ISKIP)
0100      INTEGER IDATA(1),HILIM(1),LOLIM(1),IDWN1(1),IDWN2(1),ITIME(5
0101      $IDATE(15),LABEL(9)
0102      DATA LABEL/2HFE,2HTP,2HBP,2HRE,2HST,2HCW,2HPR,2HRL,2HCL/

```



```

0103 2000  FORMAT(/,5X,"A L E R T      LOOP - ",A2)
0104 2010  FORMAT(/,5X,"D A N G E R      DIST. COL. DOWN @ ",15A2,/,5X,
0105      $"DOWN LOOP =",A2)
0106 2020  FORMAT(/,5X,"BDC99 DOWN @ ",15A2,5X,"D.L. = ",A2)
0107      CALL FTIME(IDATE)
0108      DO 5000 I=1,NDATA
0109      IF(IDATA(I).LE.LOLIM(I).OR.
0110      $IDATA(I).GE.HILIM(I)) GO TO 5010
0111      GO TO 5020
0112 5010  IF(IDWN1(I).GT.IDWN2(I)) GO TO 5030
0113      IDWN1(I)=IDWN1(I)+1
0114      IDOWN=0
0115      WRITE(LUN,2000) LABEL(I)
0116      ISKIP=ISKIP+3
0117      GO TO 5000
0118 5020  IDWN1(I)=0
0119      IDOWN=0
0120 5000  CONTINUE
0121      RETURN
0122 5030  IDOWN=1
0123      WRITE(LUN,2010) IDATE,LABEL(I)
0124      WRITE(1,2020) IDATE,LABEL(I)
0125      RETURN
0126      END
0127  C
0128  C
0129      SUBROUTINE ACQUI(LUN,NDATA,ITY1,ITY2,ICHAN,IDATA,DCDATA,
0130      $CONA,CONB,IERR)
0131      COMMON ICOM(1)
0132      DIMENSION DCDATA(1),CONA(1),CONB(1)
0133      INTEGER ICHAN(1),IDATA(1),ITY1(1),ITY2(1)
0134 2000  FORMAT(/,5X,"DIST. COL. - LOOP ",I2,"      IERR = ",I2)
0135      DO 5000 I=1,NDATA
0136      IF (ITY1(I).EQ.2) GO TO 5010
0137      CALL AIRD(1,ICHAN(I),IDATA(I),IERR)
0138      IF (IERR.EQ.1) GO TO 5020
0139      WRITE(LUN,2000) I,IERR
0140      RETURN
0141 5020  IF (IDATA(I).GE.1000.AND.IDATA(I).LE.5000) GO TO 5021
0142      IF (IDATA(I).LT.1000) IDATA(I)=1000
0143      IF (IDATA(I).GT.5000) IDATA(I)=5000
0144 5021  RDATA=0.0250*FLOAT(IDATA(I))-25.0
0145      GO TO 5030
0146 5010  IERR=1
0147      IDATA(I)=ICOM(ICHAN(I))
0148      RDATA=FLOAT(IDATA(I))
0149      GO TO 5030
0150 5030  IF (ITY2(I).EQ.2) GO TO 5040
0151      DCDATA(I)=CONA(I)*SQRT(RDATA)+CONB(I)
0152      GO TO 5000
0153 5040  DCDATA(I)=CONA(I)*RDATA+CONB(I)
0154 5000  CONTINUE
0155      RETURN
0156      END

```


B30284